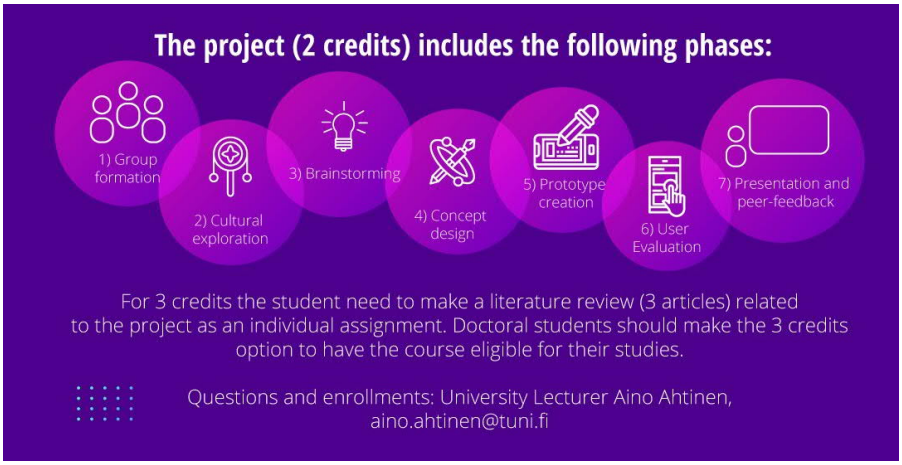
Tampereen yliopisto
Tampere University

COMP.SE.100-EN ItSE

Zoom begins soon...

at 1415 o'clock.



Tampereen yliopisto
Tampere University

COMP.SE.100-EN, 2020, course schedule v7 (11.11.2020)

week	lectures	exam	weekly exercises	project assignment (exercise work)	week
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...	35
36	Project Assignment explained		**WE1**: intro to requirements	grouping, groups to Moodle	36
37	L2: Sw Eng in general		**WE2**: Trello and agile way	group's Trello board ready with product backlog	37
38	L3: requirements		**WE3**: feasibility study and stakeholder analysis	working...	38
39	L4: basic UML diagrams		**WE4**: requirements	working...	39
40	L5: more UML diagrams	EXAM-1	**WE5**: UML diagrams - Use case	working...	40
41	L6: different sw systems	EXAM-1	**WE6**: UML diagrams - concept/entity and navigatio	deadline for 1st phase documentation and presentation	41
42	examination week		examination week	examination week	42
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations	43
44	L8: quality and testing		**WE7**: development processes	feedback group-to-group at PRP, from 1st phase	44
45	L9: project work	Forms-2	**WE8**: testing and error reporting	deadline for diagrams first versions (Moodle)	45
46	L10: project management		**WE9**: effort estimation	feedback to groups from diagrams (from assistants)	46
47	L11: open source, APIs, IPR		**WE10**: delivery contracts and terms of use	deadline for 2nd phase presentation (PRP)	47
48	L12: embedded systems, IoT		groups' final presentations	groups' final presentations	48
49	L13: recap, summary	Forms-3	- - -	g-to-g PRP feedback & final (2.) delivery of proj doc	49
50	examination week		examination week	feedback inside group, student-to-student at PRP	50
51	examination week		examination week	end of game / game over.	51
	Lectures: Wed at 1415-16.		Weekly exercises:		
			Mon 0815-10 discontinued	AUTUMN 2020 (1-2. periods)	
			Mon 1215-14	are remote/distant learning.	
			Tue 0815-10		
			Tue 1415-16		
			Wed 0815-10 discontinued .		

Remote/distant learning 2020.
No contact teaching at ItSE 2020.

Tampereen yliopisto
Tampere University

COMP.SE.100 -EN "ItSE"

Introduction to Software Engineering

2020, 1-2. periods

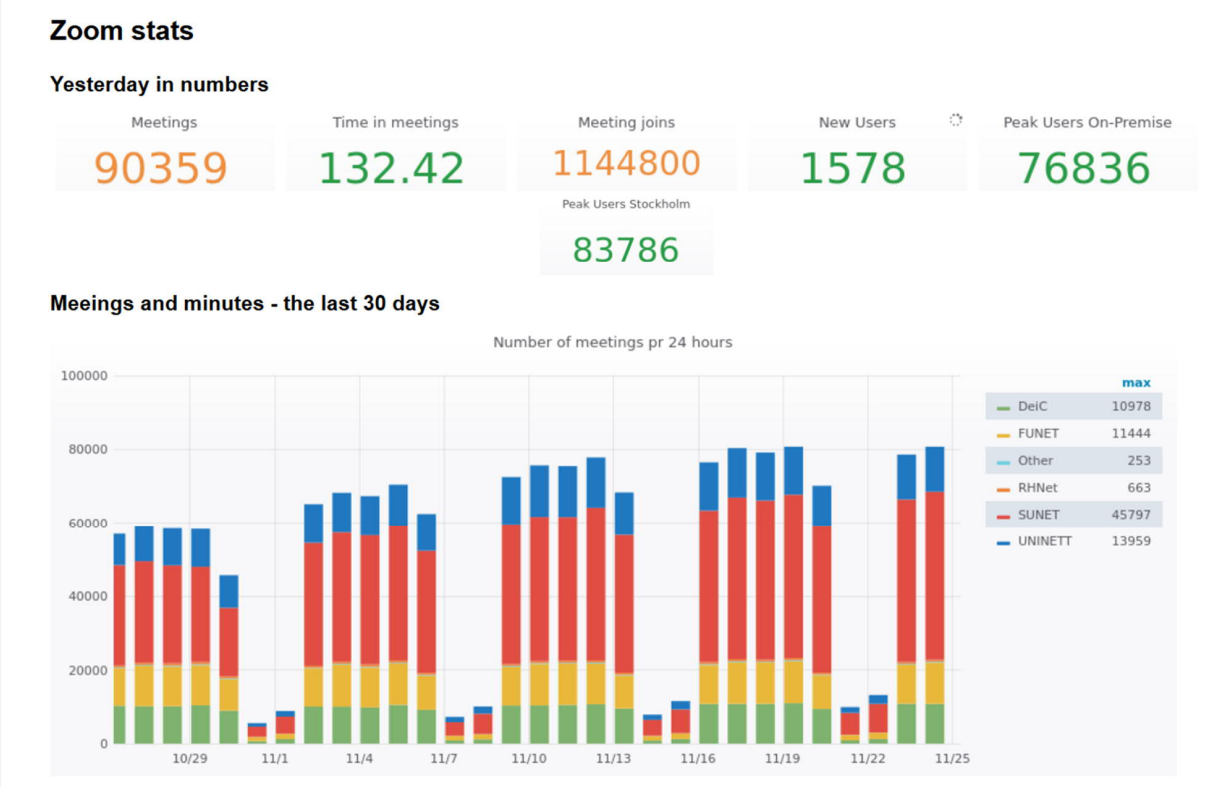
5 credit units

12-emb-ItSE-2020-v7

Tampereen yliopisto
Tampere University

Zoom statistics

FUNET = Finnish university network



Tampereen yliopisto
Tampere University

COMP.SE.100-EN (ItSE)
Introduction to Software Engneering

Lecture 12, 25.11.2020

Tensu: remember to start Zoom lecture recording, at 1415

Prefer course Moodle over SISU information.

Students are recommended to follow Moodle News/messages.

Tampereen yliopisto
Tampere University

Course contents (plan)

1. Course basics, intro
2. Sw Eng in general, overview
3. Requirements
4. Different software systems
5. Basic UML Diagrams (”Class”, Use Case, Navigation)
6. UML diagrams, in more detail
7. Life Cycle models
8. Quality and Testing
9. Project work
10. Project management
11. Open source, APIs, IPR
12. Embedded systems
13. Recap

Tampereen yliopisto
Tampere University

12. Embedded systems

- embedded systems (sw + hw)
- real-time systems (RT)
- software safety
- cyber security
- AI (artifical intelligence)
- benchmarking
- CMM / CMMI
- certificates and certification, auditing
-

Tampereen yliopisto
Tampere University

Current at course (w 48)

- WE has ended last week 47
- continue updating your Trello (kanban) boards = use at your process

- deadline for 2nd (final) presentations (week 47)
- final (2nd) presentations this week 48 (4 group slides late)
- group to group feedback at PRP (week 49)
- final delivery of req. documentation (week 49) Moodle
- peer feedback inside group at PRP (week 50)

- Third exam is changed to Forms, as covid is still here.
- EXAM 3/3 w48-49 is changed to Forms-3 exam, to be on Wednesday, 02.12.2020 starting at 1615 o'clock (week 49).

General course matters

Project assignment (exercise work)

Juanita: groups G01-G04

Aleksius: ODD groups; G05,G07,G09,G11,G13,G15,G17,G19,G21,G23,G27

Lauri: EVEN groups; G06,G08,G10,G12,G14,G16,G18,G20,G22,G24,G28

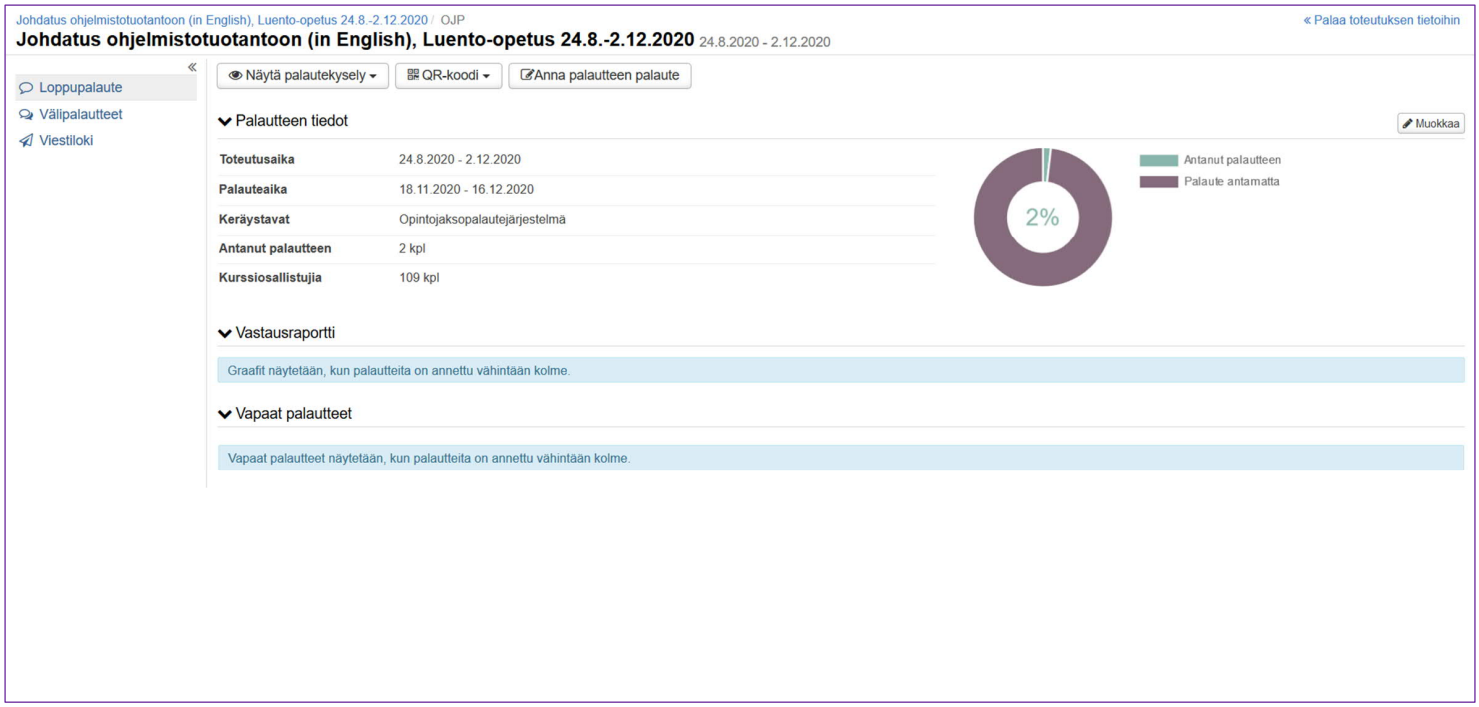
- Trello board is used as help for work division and assignment

WE attendees:

- Mon 0815-10 9, 8,10, 5, 6, 4,
- **Mon 1215-14 11,12,12,13,11,12,10, 9, 9, 6.**
- **Tue 0815-10 3, 6, 4, 6, 5, 5,10,12,10,11.**
- **Tue 1415-16 8,10, 9, 8, 5, 7,12,12,12,10.**
- Wed 0815-10 12,11, 9, 8, 7, 6,

Very small WEs are not reasonable, we discontinued two WE groups at 2nd study period.

Course feedback seems to be open 18.11.-16.12.2020



Tampereen yliopisto
Tampere University

COMP.SE.100-EN, 2020, course schedule v7 (11.11.2020)

week	lectures	exam	weekly exercises	project assignment (exercise work)	week
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...	35
36	Project Assignment explained		**WE1**: intro to requirements	grouping, groups to Moodle	36
37	L2: Sw Eng in general		**WE2**: Trello and agile way	group's Trello board ready with product backlog	37
38	L3: requirements		**WE3**: feasibility study and stakeholder analysis	working...	38
39	L4: basic UML diagrams		**WE4**: requirements	working...	39
40	L5: more UML diagrams	EXAM-1	**WE5**: UML diagrams - Use case	working...	40
41	L6: different sw systems	EXAM-1	**WE6**: UML diagrams - concept/entity and navigatio	deadline for 1st phase documentation and presentation	41
42	examination week		examination week	examination week	42
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations	43
44	L8: quality and testing		**WE7**: development processes	feedback group-to-group at PRP, from 1st phase	44
45	L9: project work	Forms-2	**WE8**: testing and error reporting	deadline for diagrams first versions (Moodle)	45
46	L10: project management		**WE9**: effort estimation	feedback to groups from diagrams (from assistants)	46
47	L11: open source, APIs, IPR		**WE10**: delivery contracts and terms of use	deadline for 2nd phase presentation (PRP)	47
48	L12: embedded systems, IoT		groups' final presentations	groups' final presentations	48
49	L13: recap, summary	Forms-3	- - -	g-to-g PRP feedback & final (2.) delivery of proj doc	49
50	examination week		examination week	feedback inside group, student-to-student at PRP	50
51	examination week		examination week	`end of game / game over.`	51
	Lectures: Wed at 1415-16.		Weekly exercises:		
			Mon 0815-10 discontinued	AUTUMN 2020 (1-2. periods)	
			Mon 1215-14	are remote/distant learning.	
			Tue 0815-10		
			Tue 1415-16		
			Wed 0815-10 discontinued .		

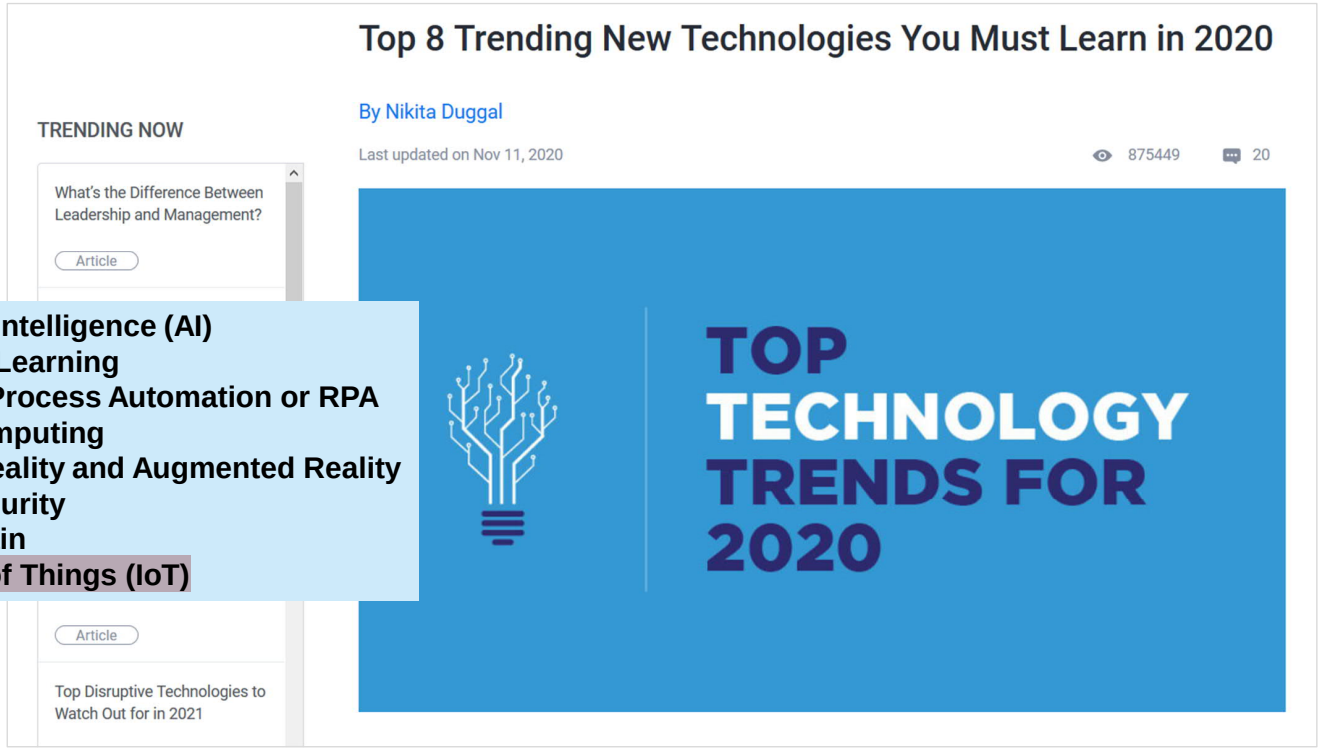
Tampereen yliopisto
Tampere University

Tampereen yliopisto
Tampere University

[https://tech.eu/brief/unikie-funding/]

Tampereen yliopisto
Tampere University

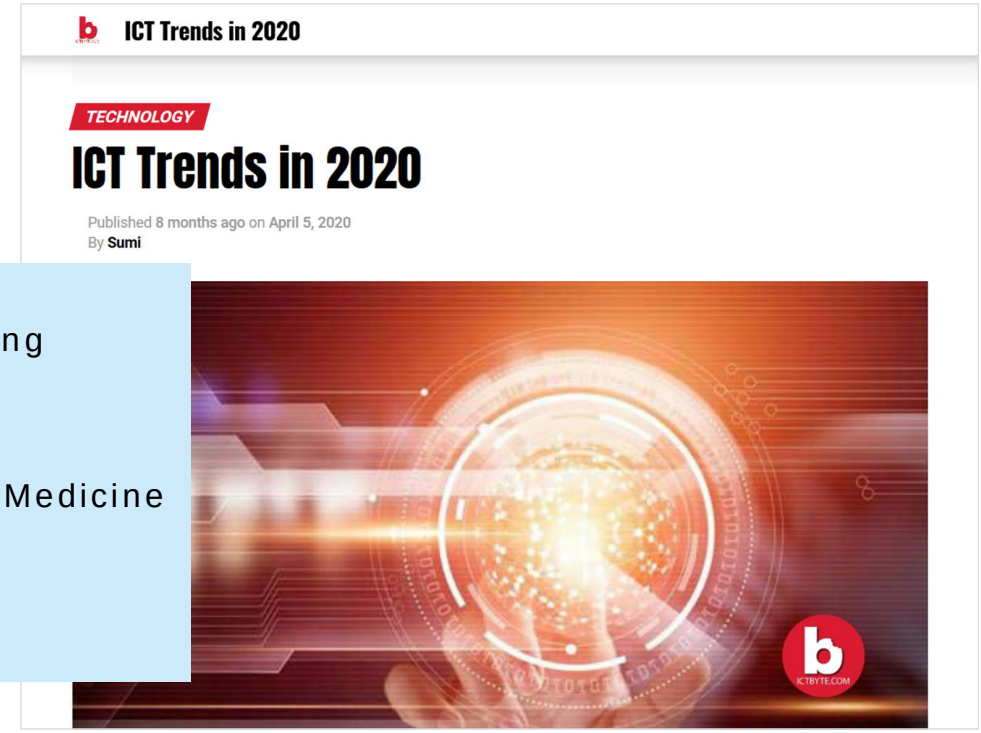
[https://www.simplilearn.com/top-technology-trends-and-jobs-article]



1. Artificial Intelligence (AI)
2. Machine Learning
3. Robotic Process Automation or RPA
4. Edge Computing
5. Virtual Reality and Augmented Reality
6. Cybersecurity
7. Blockchain
8. Internet of Things (IoT)

Tampereen yliopisto
Tampere University

[https://ictbyte.com/technology/ict-trends-in-2020/]



7 ICT Trends in 2020

- AI and Machine Learning
- 5G
- Autonomous Driving
- Predictive & Precision Medicine
- Computer Vision
- Extended Reality
- Blockchain.

Tampereen yliopisto
Tampere University

Embedded systems

Tampereen yliopisto
Tampere University

Embedded systems

Embedded systems (FI: sulautetut järjestelmät)
have software and hardware working closely together.

For example

- mobile phones
- lift/elevator
- coffee maker and other kitchen machines
- DVD player
- television
- laser printer
- aircraft.

Tampereen yliopisto
Tampere University

Embedded systems

Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke.

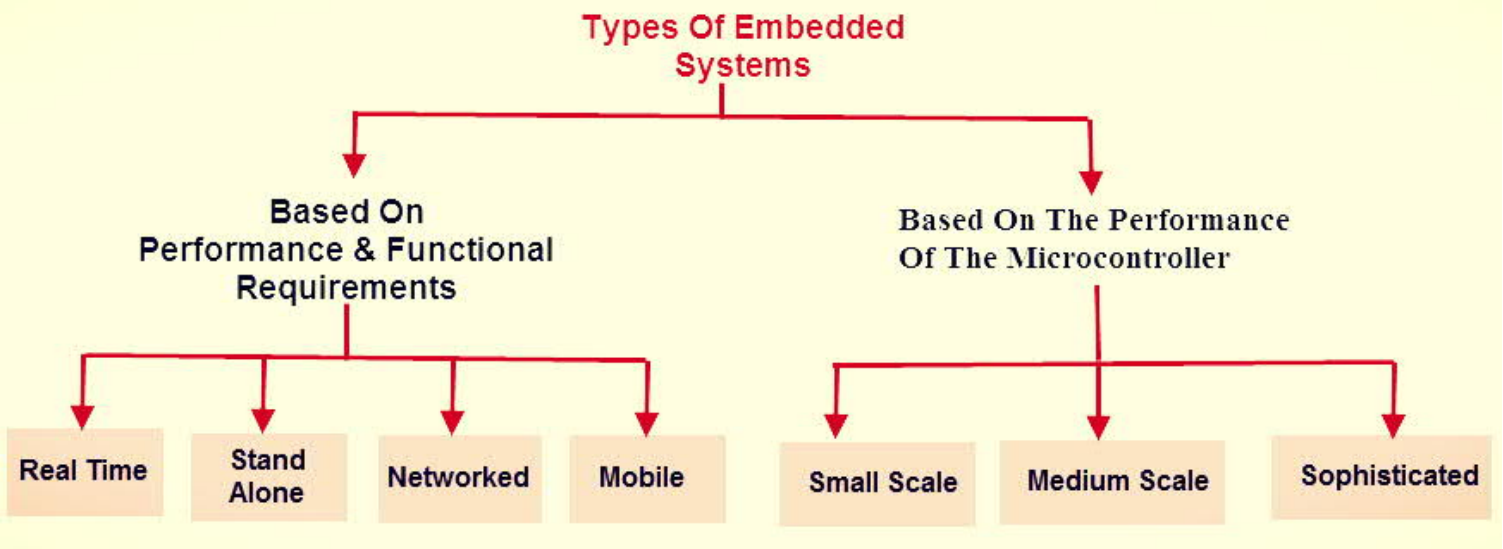
An embedded system has three components

- It has hardware.
- It has application software.
- It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small scale embedded system may not have RTOS.

So we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

[https://www.tutorialspoint.com/embedded_systems/]

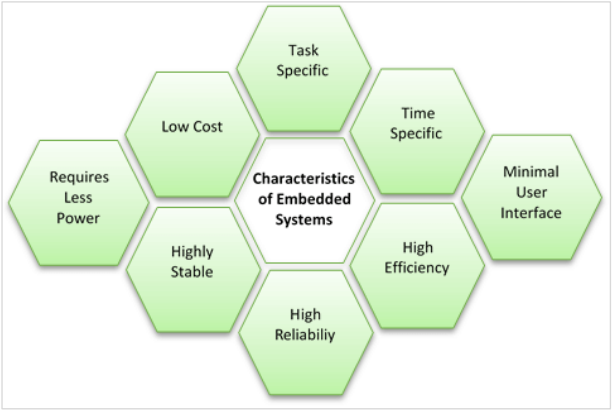
Embedded systems types



[https://www.itrelease.com/2018/07/examples-and-types-of-embedded-systems/]

Embedded system [https://www.guru99.com/embedded-systems-tutorial.html]

Embedded System is a combination of computer software and hardware which is either fixed in capability or programmable. An embedded system can be either an independent system, or it can be a part of a large system. It is mostly designed for a specific function or functions within a larger system. For example, a fire alarm is a common example of an embedded system which can sense only smoke.



Important characteristics of an embedded system:

- requires real time performance
- it should have high availability and reliability
- developed around a real-time operating system
- usually, have easy and a diskless operation, ROM boot
- designed for one specific task
- offers high reliability and stability
- needed minimal user interface
- it must be connected with peripherals to connect input and output devices
- limited memory, low cost, fewer power consumptions
- it does not need any secondary memory in computer.

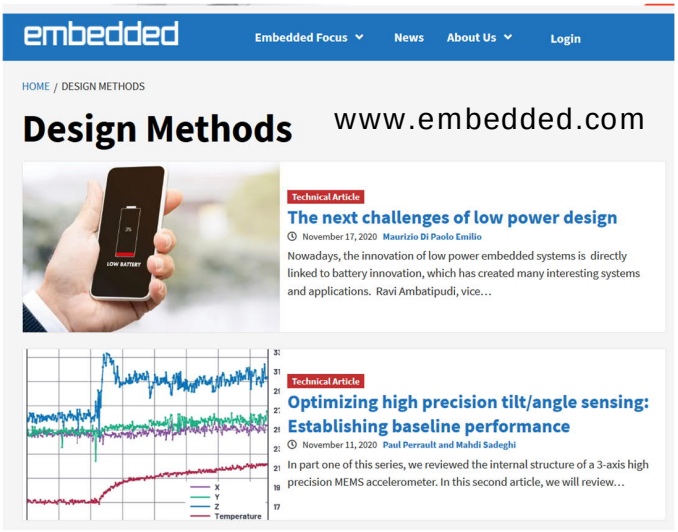
Technology evolves rapidly…

Search YouTube for videos e.g. from

- Boston Dynamics (walking robot models).

Fake and/or fun YT-videos by

- Boston dynamixx
- Bostown dynamics.



State machine diagrams are help

In embedded systems (and real-time systems, too) you should think about the order of different actions happening in the system.

State (transition) diagrams may help this a lot. There you draw possible state transitions. Sometimes you have better make sure at code level, that unwished state transitions DO NOT happen.

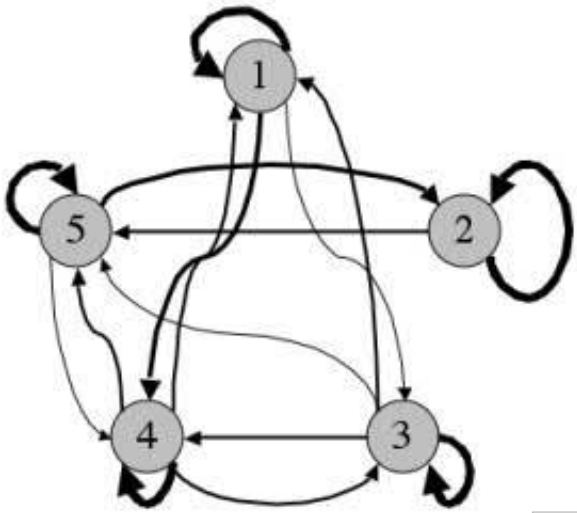
If your system is part of some larger system, remember that changes to the main system MAY affect the functionality (although it should not).

Even normal updates sometimes break something. E.g. mobile phones, copying machines, TV sets, TUNI EXAM, Zoom.

State transition diagram and matrix

[http://web.mit.edu/9.29/www/neville_jen/hmm/]

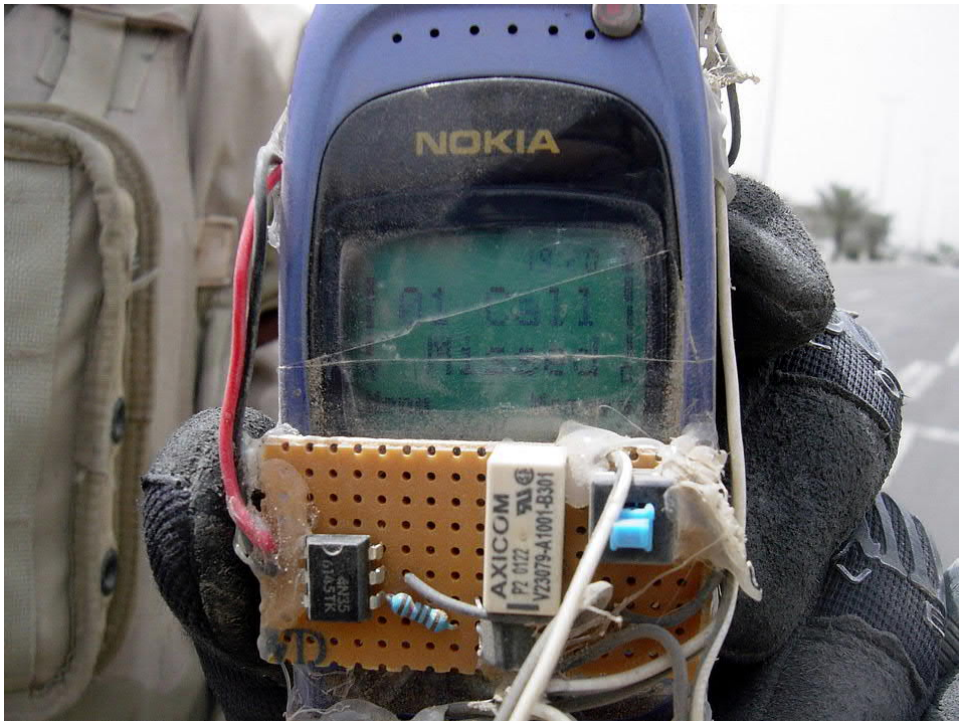
Sometimes state matrix may reveal some transitions which should be absolutely denied.



Start State \ Dest. State	1	2	3	4	5
1	.977		.001	.021	
2		.997			.003
3	.005		.992	.003	.0001
4	.008		.004	.982	.006
5		.012		.0006	.988

16 actual state transitions,
25 possible state transitions

Is this an embedded system ?



Tampereen yliopisto
Tampere University

embedded software, e.g. cars

Traditionally, in old times, embedded systems were made hardware first (as it was slow development), and then software. A common misunderstanding was that ”hardware could be made almost working, as the functionality could be fixed easily with” software. Such happened e.g. at first F-18 fighter planes.

Nowadays you have better think, plan and design those in parallel.

For car software, earlier there were problems (not anymore ?) because (self-educated) ”car engineers” who had no ”sw dev training” started to write software.

For mobile phones, you may develop

- native application for a certain operating system, if you want quick response times and special features
- general web application, which runs on some browser (also at PC), if you want ”easy” development and wide use.

Tampereen yliopisto
Tampere University

Some cross-platform mobile dev tools

Xamarin is the preferred mobile app development tool for native applications. It reuses business logic layers and data access across platforms. It is widely used to build apps for iOS, Windows, and Android app development.

Appcelerator allows developers to create apps with fewer lines of code. This app development tool supports iOS, Android, Windows, and browser-based HTML5 applications.

PhoneGap is an Open Source free to use mobile app development framework. It falls into the category of cross-platform app development. It can be used for developing a single app which works on all mobile devices.

Ionic is HTML5 mobile app development framework. It is widely used for developing hybrid mobile apps. It is a useful tool to build mobile apps using web technologies like CSS, HTML5, and SASS.

Qt cross-platform SDK. It offers cost-effective design, development, and deployment. It allows developers to deliver the best user experience across all devices.

Alpha Anywhere is a rapid mobile app development and deployment tool. It is used for building cross platform web and mobile business apps.

[https://www.guru99.com/mobile-app-development-tools.html , 2019]

Summary

SIEMENS

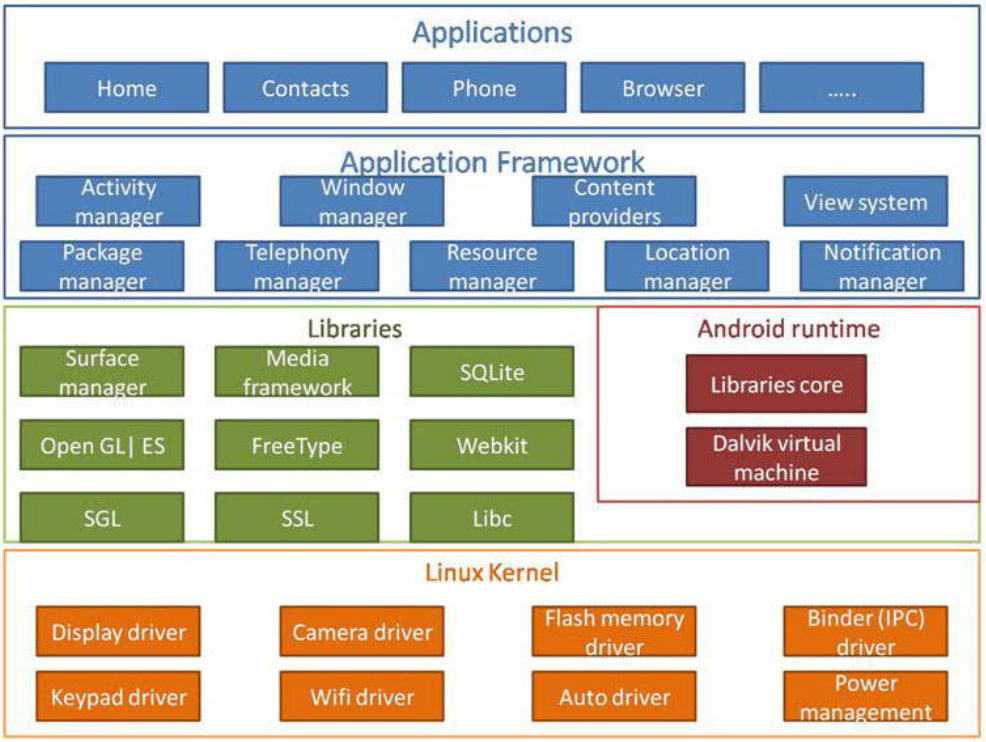
The 7 Challenges of Embedded Software Development

1. **Consolidation**
   - shift from HW to SW
   - utilization of multi-/many-core systems
   - taking into account safety and real-time requirements
2. **Decentralization**
   - flexible deployment of functionality in distributed systems
3. **Heterogeneity**
   - heterogeneous multi-/many-core architectures
   - hardware accelerators
   - cloud computing
4. **Security**
   - data privacy
   - protection against manipulation
5. **Energy management**
   - power-efficient hard- and software
6. **Programming models**
   - Development efficiency and future-proofness
   - Portability, HW-independence
   - Scalability with processing power (more cores)
7. **Migration strategies**
   - utilize parallel hardware preserving existing code bases

Seite 13 Januar 2012 Urs Gleim

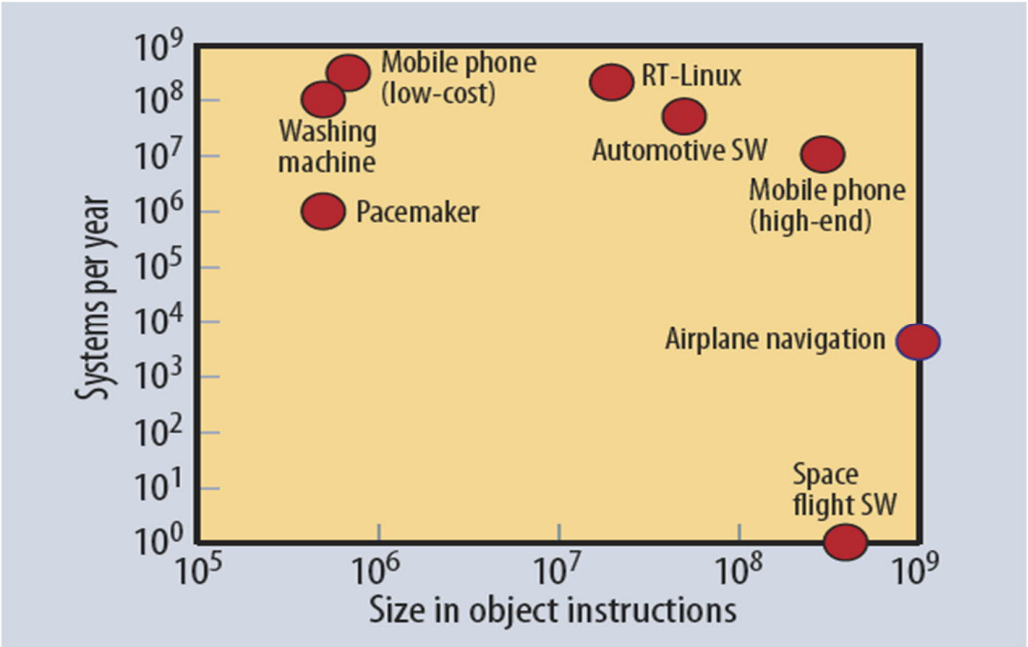
Tampereen yliopisto
Tampere University

Android OS consists of different layers of software. Each layer groups several programs and each program has its own service to provide. Together, with other applications, these layers form the OS, middleware and applications.



[http://www.learncertification.com/study-material/android-software-stack]

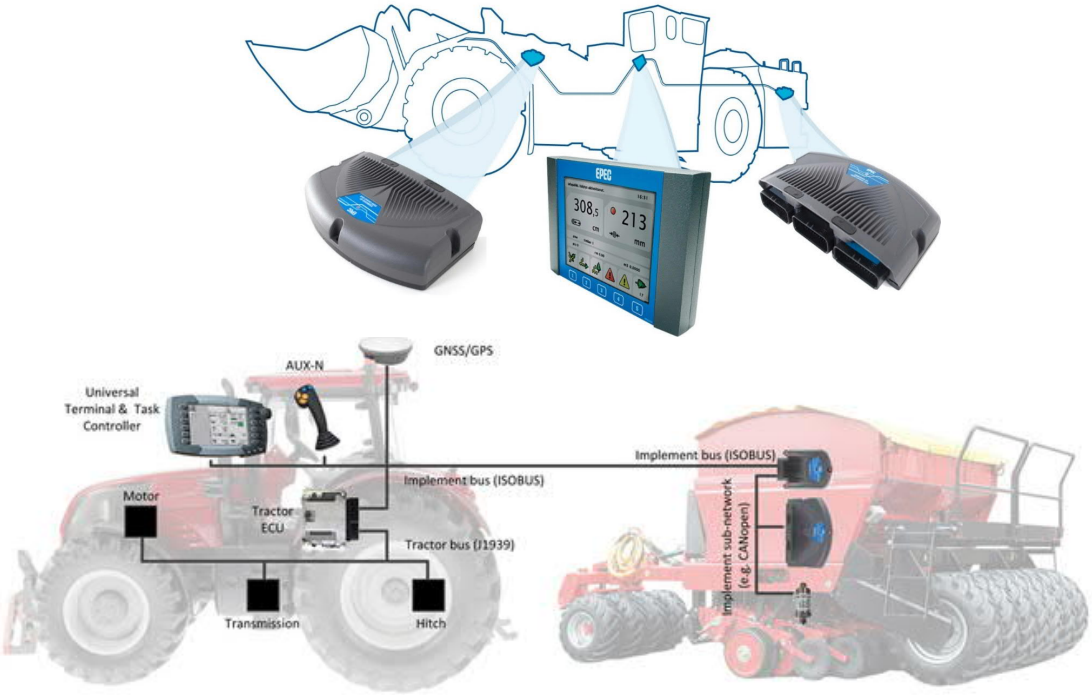
Software size and complexity, increasing



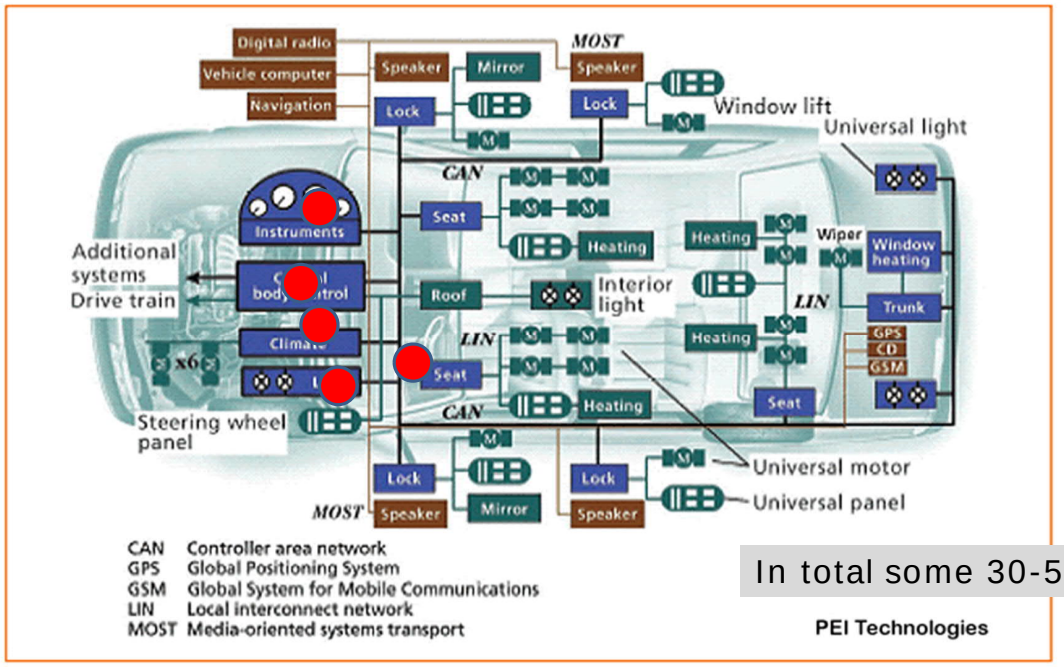
Embedded Software: Facts, Figures, and Future IEEE Computer, April 2009 (vol. 42 no. 4)

Modern car owner's pain...

Architecture (ISOBUS)

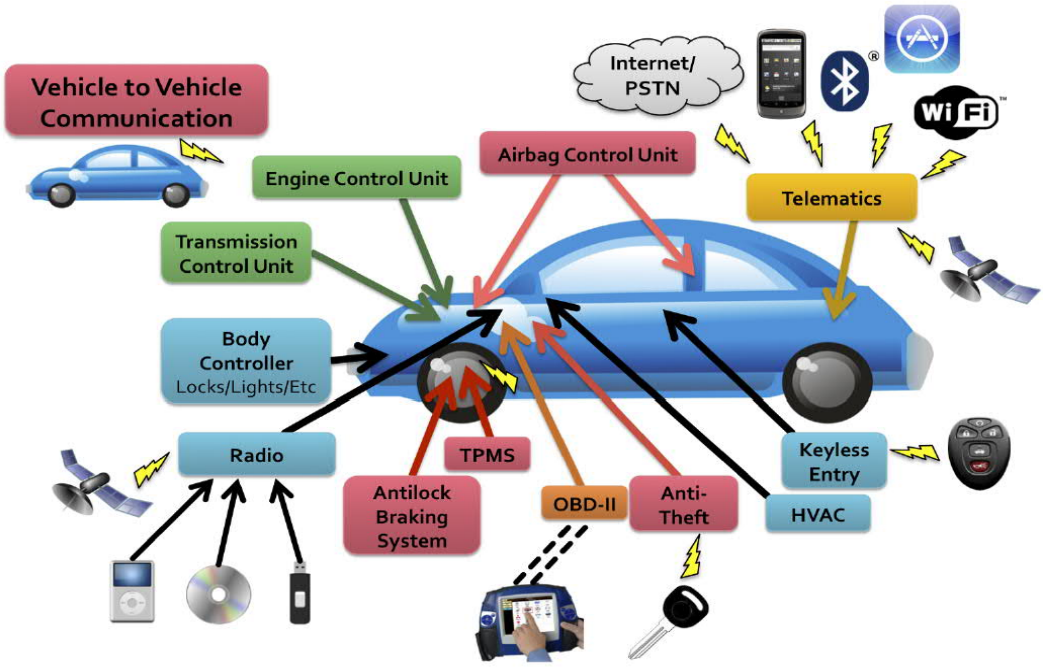


http://www.aa1car.com/library/can_systems.htm

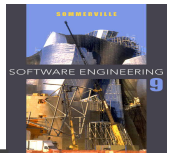


In total some 30-50 pcs

http://www.dailytech.com/Charlie+Miller+Releases+Open+Source+Car+Sabotage+Toolkit/article33308.htm



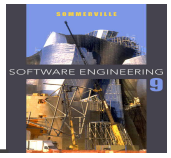
Embedded system design



- The design process for embedded systems is a systems engineering process that has to consider, in detail, the design and performance of the system hardware.
- Part of the design process may involve deciding which system capabilities are to be implemented in software and which in hardware.
- **Low-level decisions on hardware, support software and system timing must be considered early in the process.**
- These may mean that **additional software functionality, such as battery and power management**, has to be included in the system.

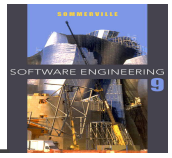
E.g. a hearth pacemaker goes at sleep mode between hearthbeats, to save energy and that way the battery lasts longer.

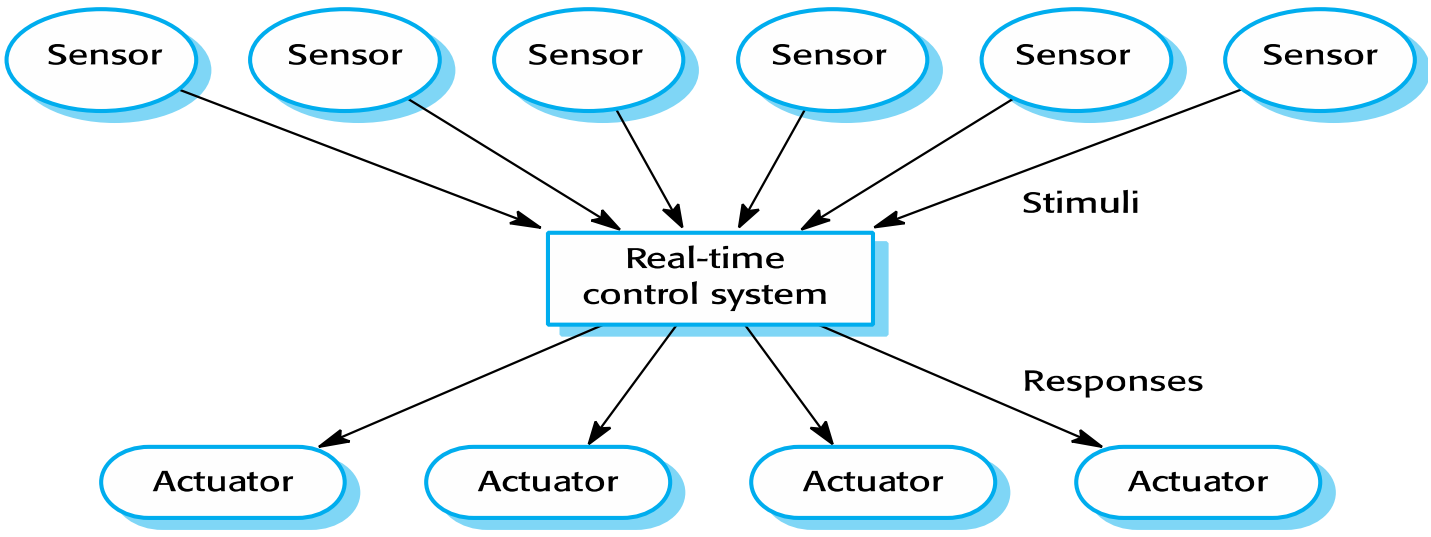
Reactive systems



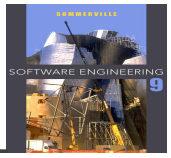
- Real-time systems are often considered to be reactive systems. Given a stimulus, the **system must produce a reaction or response within a specified time**.
- Periodic stimuli. Stimuli which occur at predictable time intervals
  - For example, a temperature sensor may be polled 10 times per second.
- Aperiodic stimuli. Stimuli which occur at unpredictable times
  - For example, a system power failure may trigger an interrupt which must be processed by the system.

A general model of an embedded real-time system



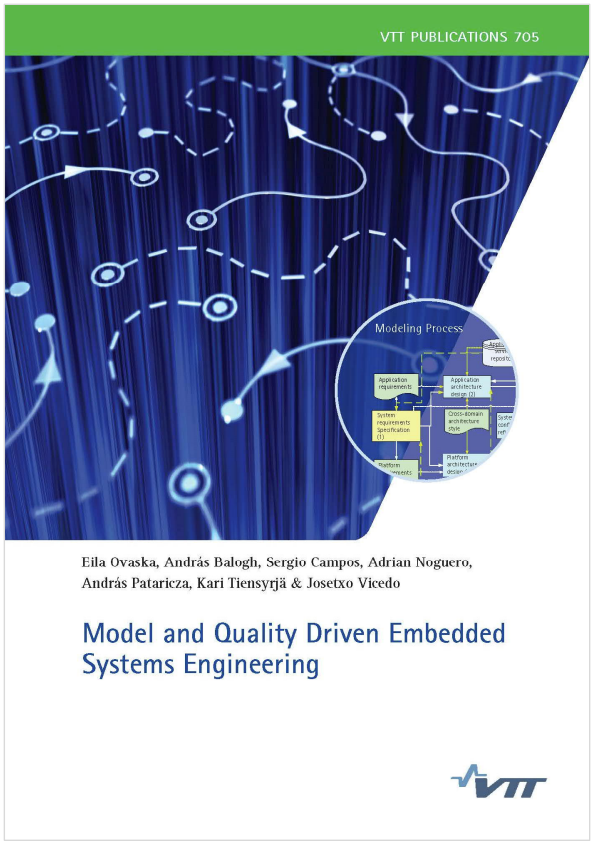


Design process activities



- ✧ Platform selection
- ✧ Stimuli/response identification
- ✧ Timing analysis
- ✧ Process design
- ✧ Algorithm design
- ✧ Data design
- ✧ Process scheduling.

Tampereen yliopisto
Tampere University



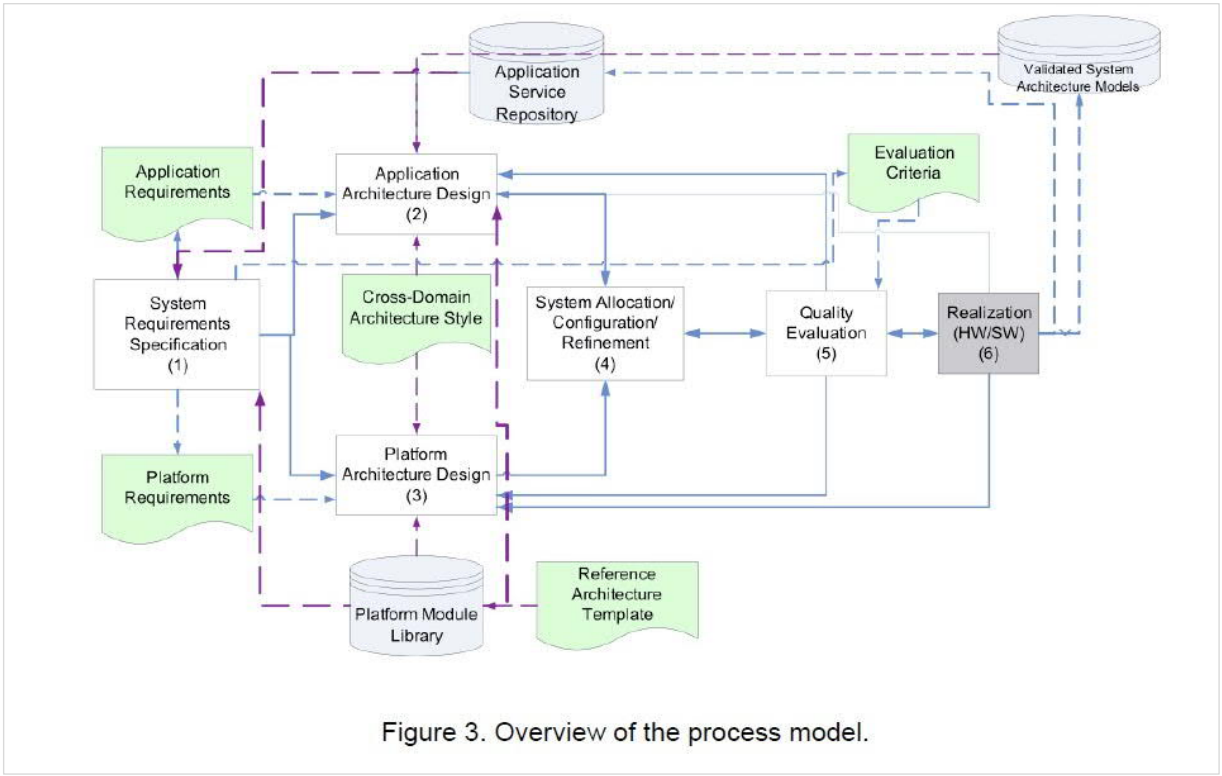
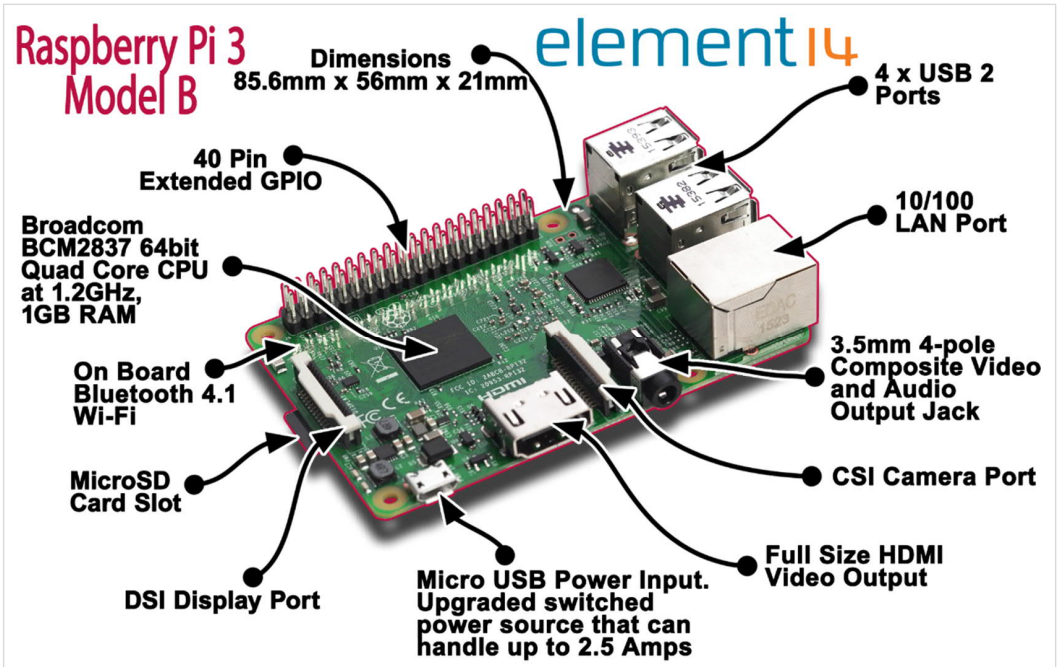


Figure 3. Overview of the process model.

There are many (and cheap) micro-computers



At old times many designers thought software can correct the hardware design flaws (not planned well).

University of Toronto — Department of Computer Science

Bugs in the Space Program:
The Role of Software in Systems Failure

Prof. Steve Easterbrook

Dept of Computer Science,
University of Toronto

http://www.cs.toronto.edu/~sme

 1

University of Toronto — Department of Computer Science

Key ideas

Complex systems rely more and more on software as the "glue"

→ Is software just like any other component?

→ Are there additional risks when we choose to allocate system functions to software?

→ Are systems engineering and software engineering similar disciplines?

University of Toronto — Department of Computer Science

Why is software special?

→ Software is invisible, intangible, abstract
- Software alone is useless - its purpose is to configure some hardware to do something

→ Software doesn't obey the laws of physics
- Behaviour explained by discrete math, rather than continuous math

→ Software has no repeated components
- Hence more complex for its "size" than other designed artifacts

→ Software never wears out
- …statistical reliability measures don't apply

→ Software can be replicated perfectly
- …no manufacturing variability

→ Software is not manufactured
- …so can be re-designed even after deployment

 5

Bugs in the space program

University of Toronto — Department of Computer Science

Factor	STS 51L	Ariane 501	Path-finder	MCO	MPL	STS 107
Didn't test to spec		●		●	●	
Insufficient test data	●	●			●	●
Tested "wrong" system		●			●	
No regression test					●	
Lack of integration testing		●		●		
System changed after testing					●	?
Requirement not implemented		?		●	●	
Lack of diagnostic data during operation			●	●	●	●
System deployed before ready	●			?	?	
Didn't use problem reporting system	●		●	●	●	
Didn't track problems properly	●	●	●	●	●	●
Didn't investigate anomalies	●		●	●		●
Poor communication between teams	●	●	●	●	●	●
Insufficient staffing	●			●	●	
Failure to adjust budget and schedule	●			●	●	
Inexperienced managers	?			●	●	
Commercial pressures took priority	●	●		●	●	●
'Redundant' design not really redundant	●	●				
Lack of expertise at inspections		●		●	●	
Different team maintains software				●	●	
Reused code w/o checking assumptions		●				

© Steve Easterbrook, 2005 — 30

STS = space shuttle

MCO = mars climate orbiter

MPL = mars polar lander

Common problems in embedded sw dev

Testing is difficult
- hardware may not be available (made) in the beginning
- reactive system
- parts are depending on the whole system
- perhaps not visible signs of software functionality outside system
- testing is difficult, no suitable tools
- subcontractors have difficulties in testing (only part of system).

Testing solutions, not good practice
- self-made test stubs
- reggression testing is difficult
- use of emulators and simulators, as no actual hw available (yet).

Emulator = software emulates hardware, e.g. in mobile phone development. Emulator is never 100 % same as actual hardware.
Simulator = software and some hardware parts, e.g. user interface panel.

Tampereen yliopisto
Tampere University

Emulator vs. simulator

[https://www.browserstack.com/guide/testing-on-emulators-simulators-real-devices-comparison]

An emulator is a software that mimics the hardware and software of the target device on your computer. They do this by translating the ISA (Instruction Set Architecture) of the target device to the one used by the computer you are using to conduct testing using binary translation.

By translating the ISA of the target mobile device into your computer, you can mimic the way your target device works, forming a virtual environment for testing.

However, these near-native capabilities of the target mobile device, that enable you to adjust the physical sensors, geolocation, etc. come at the cost of latency.

A simulator is a software that helps your computer run certain programs built for a different Operating System. They are mostly meant for iPhone and iPad devices, unlike Android devices that can be emulated easily.

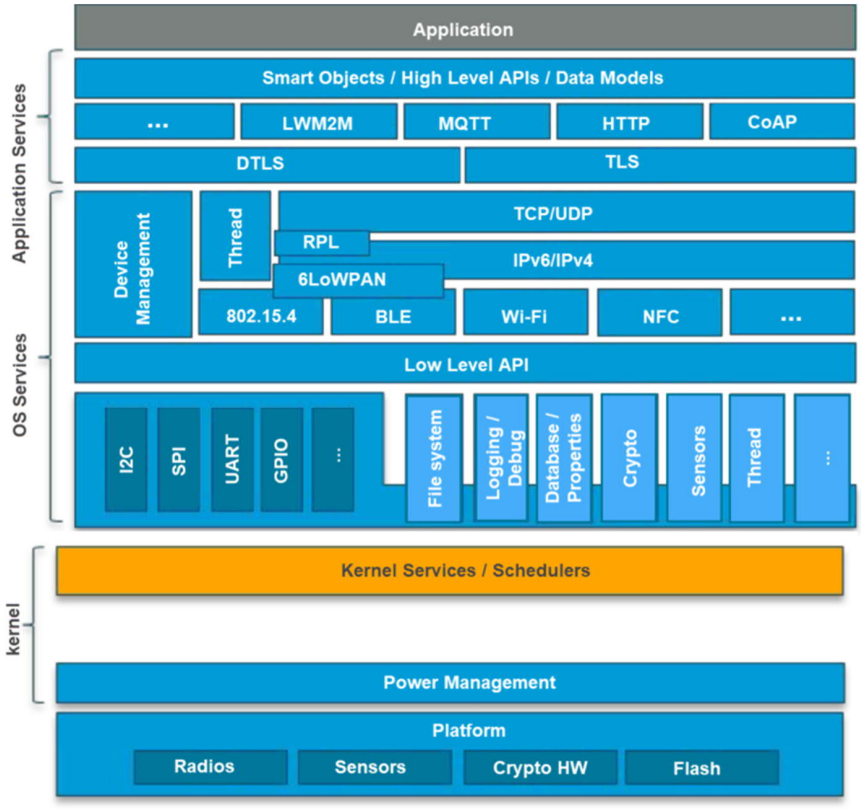
The iOS simulators mimic iOS and run the required application inside it, by sitting on top of the computer's Operating System. But to run the iOS simulator, one needs to work on the macOS only, as it needs Apple's native Cocoa API. This Cocoa API is essential for the GUI, runtime and many other operations.

Simulators unlike emulators, do not mimic hardware. Thus one cannot investigate certain functionalities like battery usage, cellular interrupts, etc. while using simulators for testing.

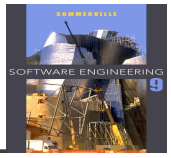
Tampereen yliopisto
Tampere University

RT, Real-time systems

Zephyr, an Open Source RTOS for IoT

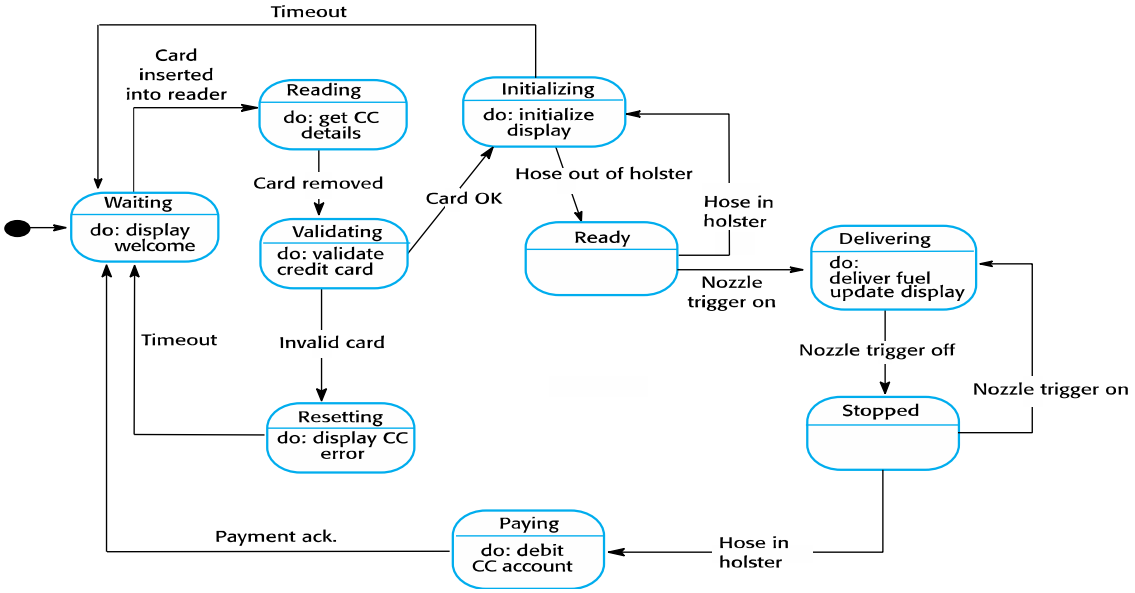


Real-time system modelling



- The effect of a stimulus in a real-time system may trigger a transition from one state to another.
- **State models are therefore often used to describe embedded real-time systems.**
- UML state diagrams may be used to show the states and state transitions in a real-time system.

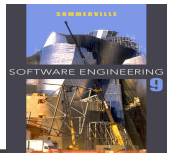
State machine model of a petrol (gas) pump



Sequence of actions in real-time pump control system

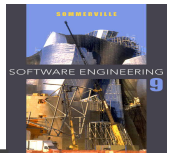
- The buyer inserts a credit card into a card reader built into the pump.
- Removal of the card triggers a transition to a Validating state where the card is validated.
- If the card is valid, the system initializes the pump and, when the fuel hose is removed from its holster, transitions to the Delivering state.
- After the fuel delivery is complete and the hose replaced in its holster, the system moves to a Paying state.
- After payment, the pump software returns to the Waiting state

Key points

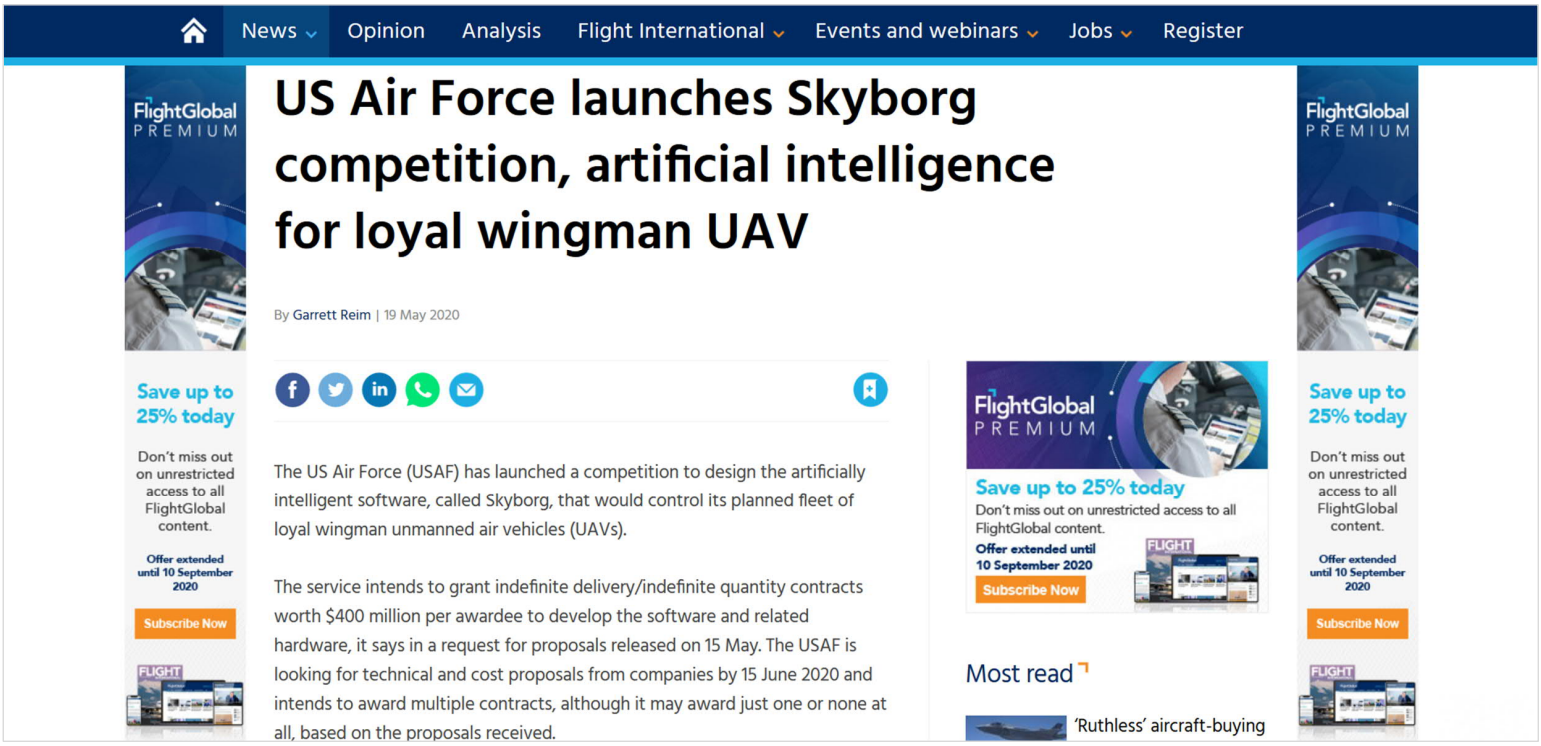


- **An embedded software system is part of a hardware/software system that reacts to events in its environment**. The software is ‘embedded’ in the hardware. Embedded systems are normally real-time systems.
- **A real-time system is a software system that must respond to events in real time**. System correctness does not just depend on the results it produces, but also on the time when these results are produced.
- Real-time systems are usually implemented as a set of communicating processes that react to stimuli to produce responses.
- **State models are an important design representation** for embedded real-time systems. They are used to show how the system reacts to its environment as events trigger changes of state in the system.

Key points



- There are several standard patterns that can be observed in different types of embedded system. These include a pattern for monitoring the system’s environment for adverse events, a pattern for actuator control and a data-processing pattern.
- **Designers of real-time systems have to do a timing analysis**, which is driven by the deadlines for processing and responding to stimuli. They have to decide how often each process in the system should run and the expected and worst-case execution time for processes.
- **A real-time operating system is responsible for process and resource management**. It always includes a **scheduler**, which is the component responsible for deciding which process should be scheduled for execution.



News Opinion Analysis Flight International Events and webinars Jobs Register

US Air Force launches Skyborg competition, artificial intelligence for loyal wingman UAV

By Garrett Reim | 19 May 2020

The US Air Force (USAF) has launched a competition to design the artificially intelligent software, called Skyborg, that would control its planned fleet of loyal wingman unmanned air vehicles (UAVs).

The service intends to grant indefinite delivery/indefinite quantity contracts worth $400 million per awardee to develop the software and related hardware, it says in a request for proposals released on 15 May. The USAF is looking for technical and cost proposals from companies by 15 June 2020 and intends to award multiple contracts, although it may award just one or none at all, based on the proposals received.

Tampereen yliopisto
Tampere University

https://www.youtube.com/watch?v=iJpeWAxk2So

Boeing Unveils First Loyal Wingman Aircraft
542,177 views • May 4, 2020

Is this the first step to Skynet? Boeing this week has unveiled an autonomous fighter jet for the Australian air force that can fly in formation with manned planes to play a role the company calls “Loyal Wingman.”

Safety

Software safety

- ”mission-critical” and ”safety-critical” software systems (space, aviation, marine, medical, military, law enforcement,…) have their own branch of safety standards

- software should do everything that is in requirements
- software should not do anything it is not supposed to do

With state diagrams, state transition matrix may be a good way to find out unwanted transitions.

So make sure, check, and double-check, make code so that certain unwanted actions CAN NOT happen.

Therac-25

Between June 1985 and January 1987, a software-controlled radiation therapy machine called the Therac-25 massively overdosed six people, resulting in serious injury and deaths. A widely cited paper published in 1993 detailed the causes of these accidents.[1,2] It wasn't an anomaly but simply the first of many, as these types of radiation-overdose accidents continue today. In 2010, a series of articles were published describing related accidents in radiation therapy.[3] Many more such events have occurred since then. Indeed, the problems aren't limited to medical linear accelerators.

Many lessons should have been learned from the Therac-25 events, most of which are generalizable to almost every industry that employs safety-critical devices. In this article, I examine each of the identified factors in these accidents to determine what progress has been made in the past 30 years. In the interest of space, the specifics of the Therac-25 events aren't detailed—instead, I focus on the state of practice 30 years after the events.

[https://www.computer.org/csdl/magazine/co/2017/11/mco2017110008/13rRUxAStVR]

Overconfidence in Software

There's still widespread belief that software doesn't fail, unlike the hardware devices it replaces. In safety-critical systems today—including medical devices, aircraft, nuclear power plants, and weapons systems—standard hardware backups, interlocks, and other safety devices are often replaced by software. Where hardware backups are still used, they're often controlled by software.

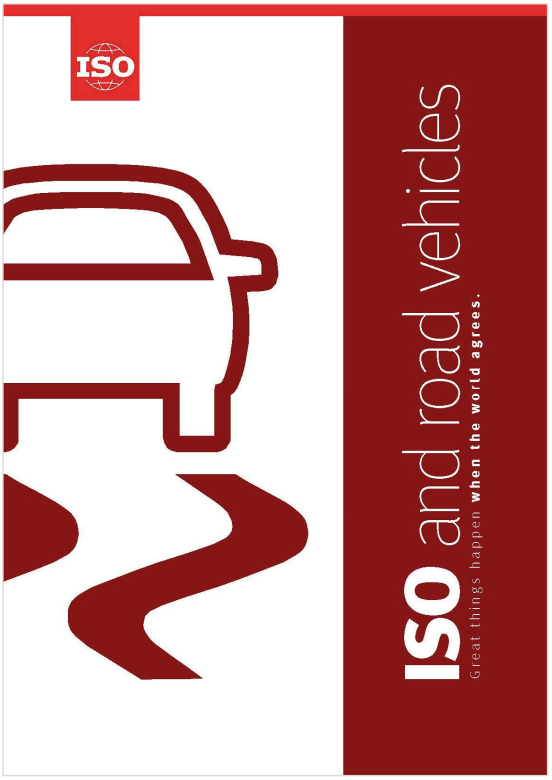
Therac-25 (1985-87)

The Blame:

The general consensus is that the Atomic Energy of Canada Limited is to blame. There was only *one* person programming the code for this system and he largely did all the testing. The machine was tested for only 2700 hours of use, but for code which controls such a critical machine, many more hours should have been put in to the testing phase. Also Therac-25 was tested as a whole machine rather then in separate modules. Testing in separate modules would have discovered many of the bugs. Also, if the AECL believed that there were problems with the Therac-25 right after the first incident then it is possible that most of the 5 other incidents could have been avoided and possibly the 3 fatalities.

http://users.csc.calpoly.edu/~csturner/courses/300/Therac25Investigation.pdf

The increasing complexity of systems and the introduction and expansion of standards, codes and certification requirements has led to a cumulative demand for functional safety analyses in the high-tech automotive industry.



ISO 26262 Road vehicles — Functional safety

Is ISO 26262 just about software development?

ISO 26262 covers the entire product development lifecycle of electrical / electronic automotive products. The standard is composed of 10 parts, as shown below:

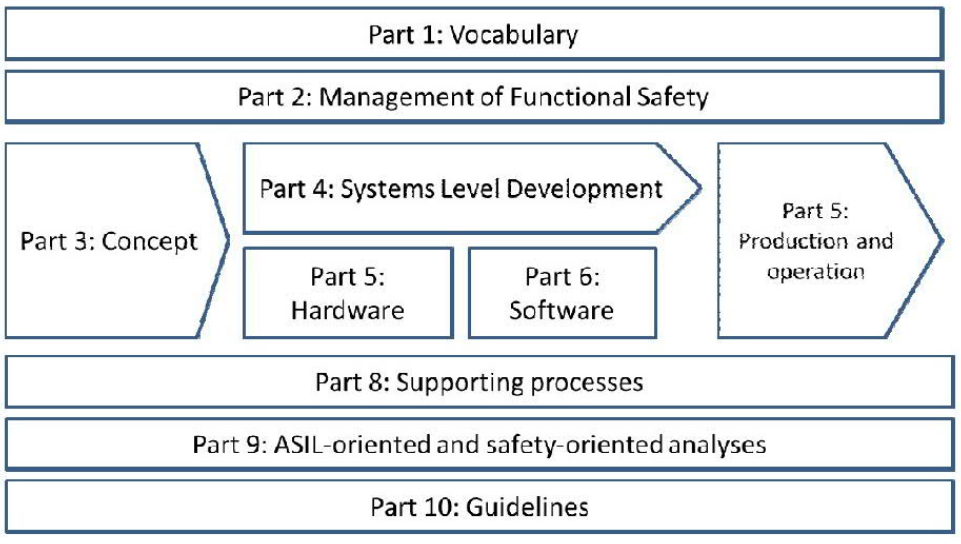
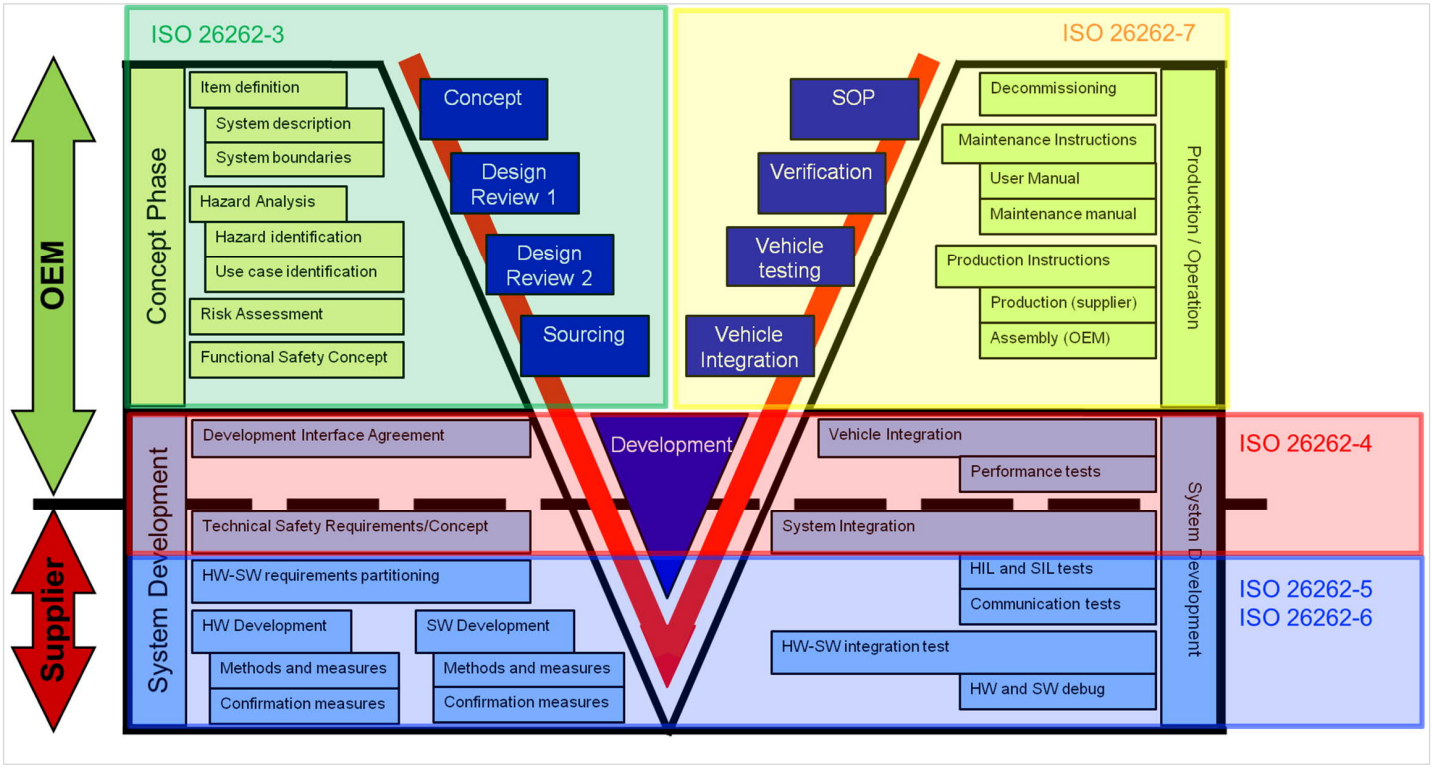


Figure 1 - The parts of ISO 26262

[www.feabhas.com]



[https://www.berns-engineers.com/be3/en/safety-engineering/functional-safety-automotive]

Tampereen yliopisto
Tampere University

DEPARTMENT OF DEFENSE

JOINT SOFTWARE SYSTEMS SAFETY ENGINEERING HANDBOOK

DEVELOPED BY THE JOINT SOFTWARE SYSTEMS SAFETY ENGINEERING WORKGROUP

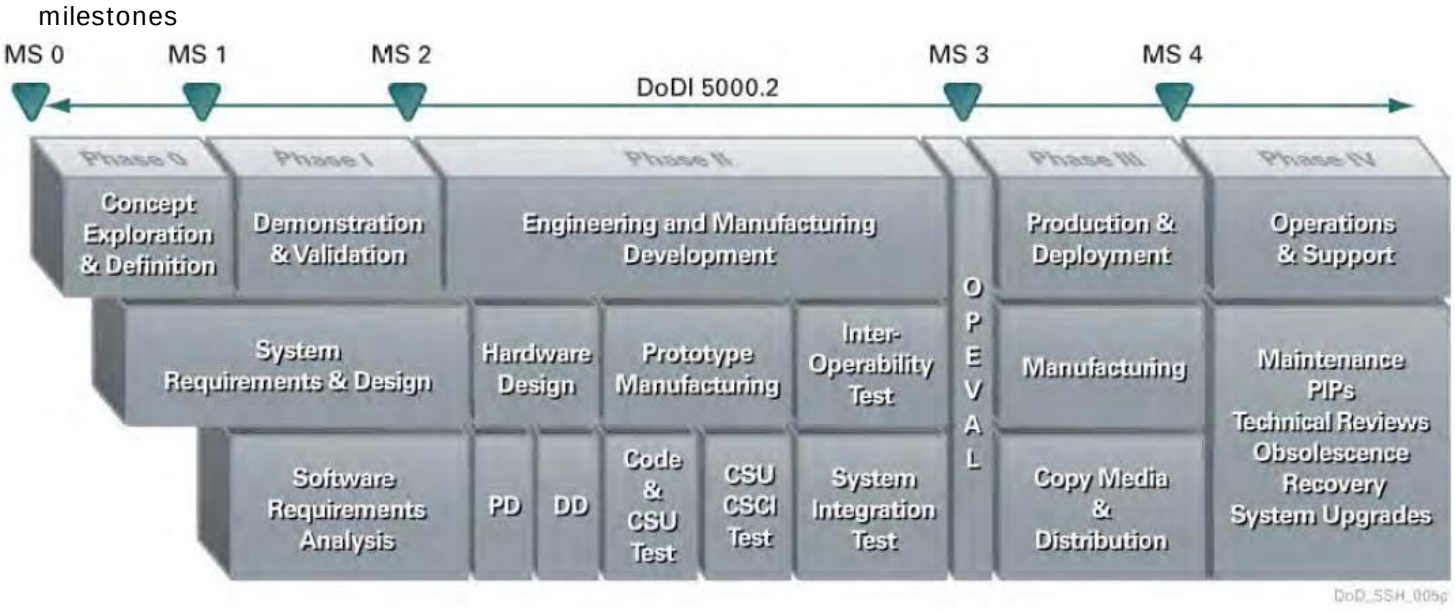
Original published December 1999
Version 1.0 Published August 27, 2010

Naval Ordnance Safety and Security Activity
3817 Strauss Avenue, Building D-323
Indian Head, MD 20640-5555

Prepared for:
Director of Defense Research and Engineering

Distribution Statement A
Approved for public release; distribution is unlimited.

Figure 2-8: Integration of Engineering Personnel and Processes



PD = program design, DD = detailed design, CSU = computer software unit, CSCI = computer software configuration item.

Figure 2-4: Relationship of Software to the Hardware Development Lifecycle

[Software System Safety Engineering Handbook, 2010]

Driver Errors, Overreliance on Automation, Lack of Safeguards, Led to Fatal Tesla Crash

9/12/2017

WASHINGTON (Sept. 12, 2017) — The National Transportation Safety Board determined Tuesday that a truck driver's failure to yield the right of way and a car driver's inattention due to overreliance on vehicle automation are the probable cause of the fatal May 7, 2016, crash near Williston, Florida.

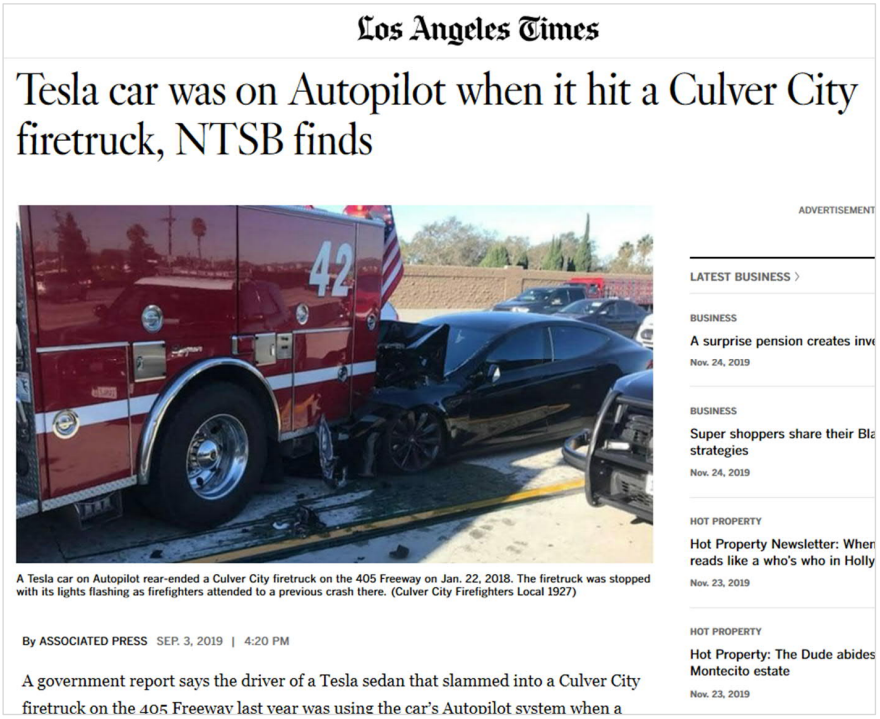
The NTSB also determined the operational design of the Tesla's vehicle automation permitted the car driver's overreliance on the automation, noting its design allowed prolonged disengagement from the driving task and enabled the driver to use it in ways inconsistent with manufacturer guidance and warnings.

As a result of its investigation the NTSB issued seven new safety recommendations and reiterated two previously issued safety recommendations.

"Neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky, so the brake was not applied," Tesla said.

Business Insider, Cadie Thompson
Jun 20, 2017, 6:06 PM

Somebody has set up a web page
https://www.tesladeaths.com/



Los Angeles Times

Tesla car was on Autopilot when it hit a Culver City firetruck, NTSB finds

A Tesla car on Autopilot rear-ended a Culver City firetruck on the 405 Freeway on Jan. 22, 2018. The firetruck was stopped with its lights flashing as firefighters attended to a previous crash there. (Culver City Firefighters Local 1927)

By ASSOCIATED PRESS SEP. 3, 2019 | 4:20 PM

A government report says the driver of a Tesla sedan that slammed into a Culver City firetruck on the 405 Freeway last year was using the car's Autopilot system when a

ADVERTISEMENT

LATEST BUSINESS

BUSINESS
A surprise pension creates inv
Nov. 24, 2019

BUSINESS
Super shoppers share their Bl
strategies
Nov. 24, 2019

HOT PROPERTY
Hot Property Newsletter: Whe
reads like a who's who in Holly
Nov. 23, 2019

HOT PROPERTY
Hot Property: The Dude abides
Montecito estate
Nov. 23, 2019

Uber-car accident 18.03.2018

A self-driving Uber Volvo XC90 SUV killed 49-year-old Elaine Herzberg as she walked her bicycle across a street in Tempe, Arizona, Sunday night, according to the Tempe Police Department.

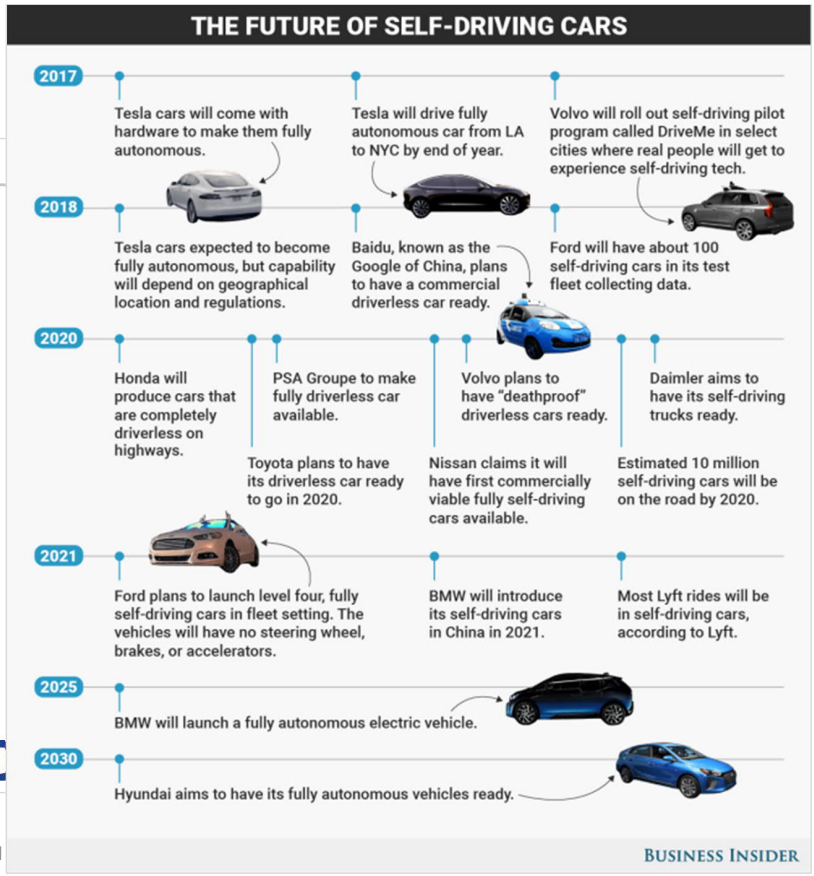
Uber's self-driving car showed no signs of slowing before fatal crash, police say

The vehicle was traveling at 40 mph

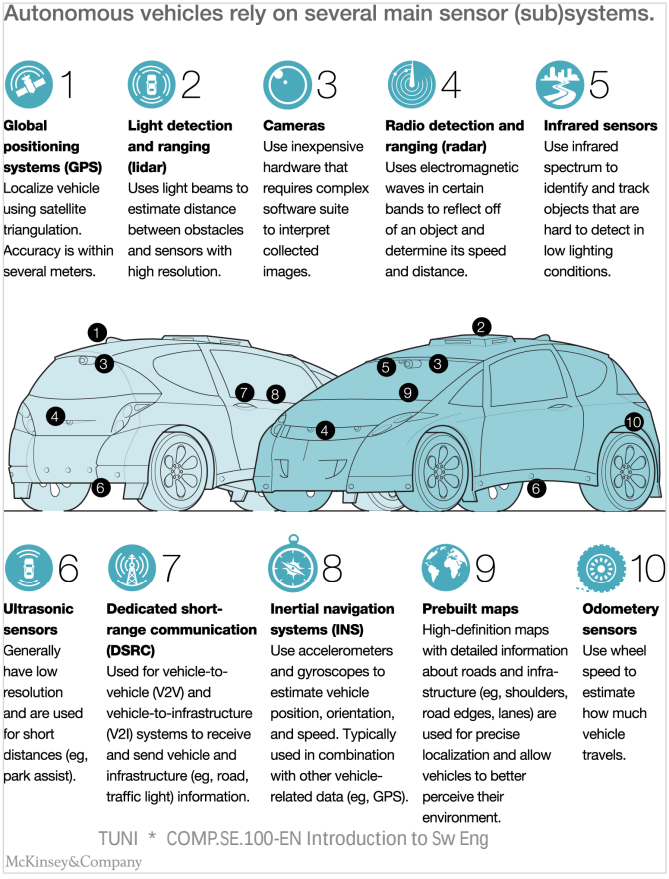
ARIZONA · 1 day ago

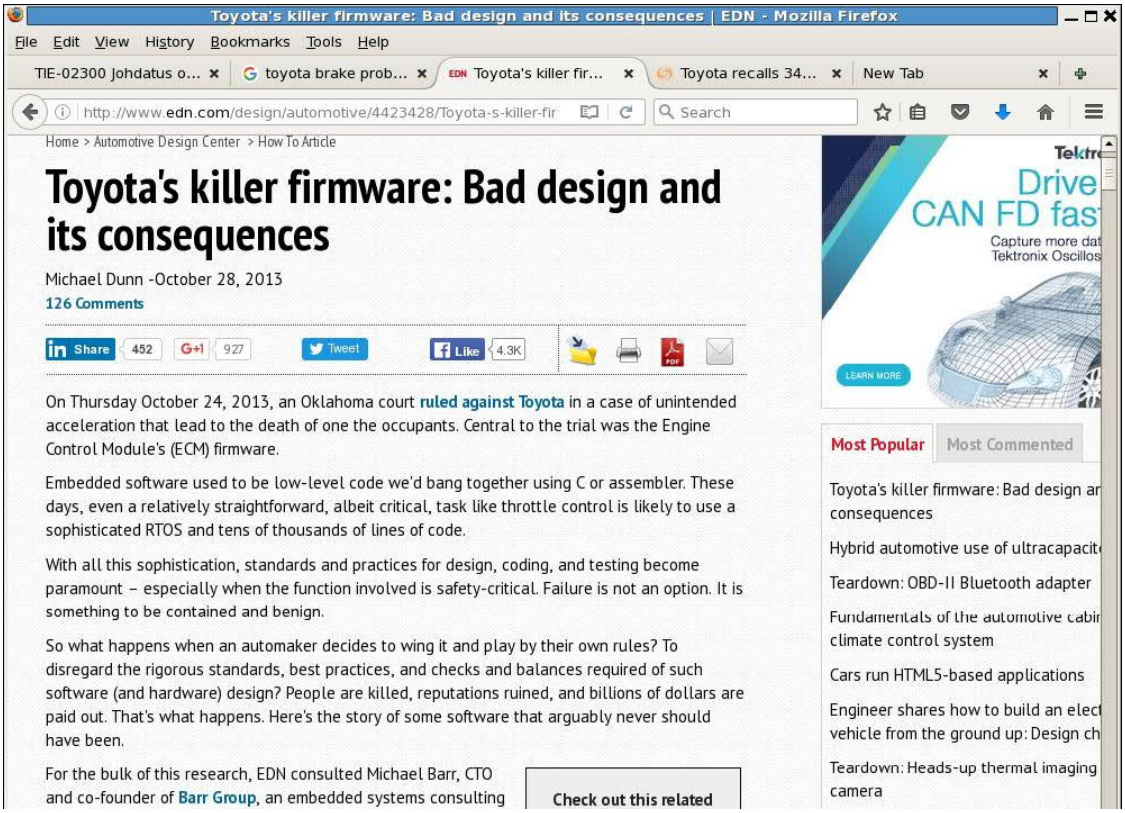
Self-driving Uber car kills Arizona pedestrian, police say

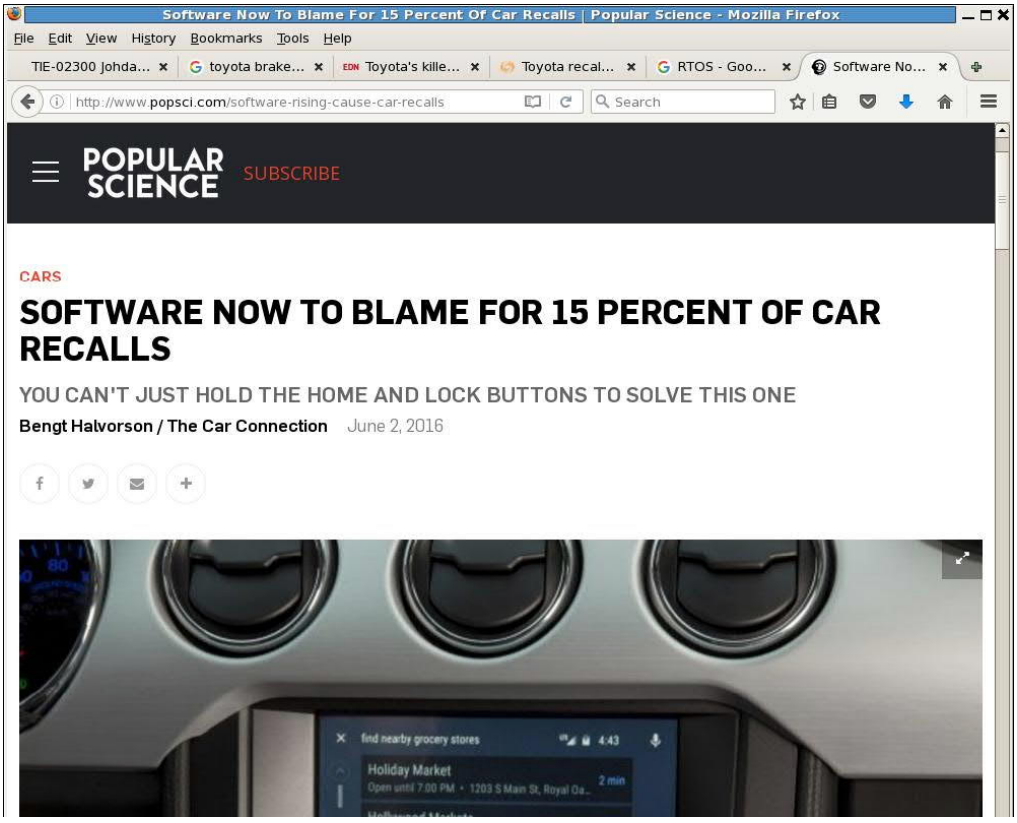
Is this developer's (coder's) fault ??



Example vehicle sensor systems




Toyota's killer firmware: Bad design and its consequences | EDN - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.edn.com/design/automotive/4423428/Toyota-s-killer-fir
Search
Home > Automotive Design Center > How To Article
Toyota's killer firmware: Bad design and its consequences
Michael Dunn -October 28, 2013
126 Comments
Share 452 G+1 927 Tweet Like 4.3K
On Thursday October 24, 2013, an Oklahoma court ruled against Toyota in a case of unintended acceleration that lead to the death of one the occupants. Central to the trial was the Engine Control Module's (ECM) firmware.
Embedded software used to be low-level code we'd bang together using C or assembler. These days, even a relatively straightforward, albeit critical, task like throttle control is likely to use a sophisticated RTOS and tens of thousands of lines of code.
With all this sophistication, standards and practices for design, coding, and testing become paramount – especially when the function involved is safety-critical. Failure is not an option. It is something to be contained and benign.
So what happens when an automaker decides to wing it and play by their own rules? To disregard the rigorous standards, best practices, and checks and balances required of such software (and hardware) design? People are killed, reputations ruined, and billions of dollars are paid out. That's what happens. Here's the story of some software that arguably never should have been.
For the bulk of this research, EDN consulted Michael Barr, CTO and co-founder of Barr Group, an embedded systems consulting
Check out this related
Drive
CAN FD fast
Capture more data
Tektronix Oscillos
LEARN MORE
Most Popular
Most Commented
Toyota's killer firmware: Bad design ar consequences
Hybrid automotive use of ultracapacit
Teardown: OBD-II Bluetooth adapter
Fundamentals of the automotive cabi climate control system
Cars run HTML5-based applications
Engineer shares how to build an elect vehicle from the ground up: Design ch
Teardown: Heads-up thermal imaging camera


Software Now To Blame For 15 Percent Of Car Recalls | Popular Science - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.popsci.com/software-rising-cause-car-recalls
Search
POPULAR SCIENCE
SUBSCRIBE
CARS
SOFTWARE NOW TO BLAME FOR 15 PERCENT OF CAR RECALLS
YOU CAN'T JUST HOLD THE HOME AND LOCK BUTTONS TO SOLVE THIS ONE
Bengt Halvorson / The Car Connection June 2, 2016
find nearby grocery stores
Holiday Market

F-22 Raptors' systems crash mid-flight over Pacific

Lockheed's shiny new F-22 Raptor stealth fighters may have owned a few war games, but crossing the International Date Line left them as helpless as a carrot in a rabbit trap, with multiple system crashes causing an emergency detour en route from Hawaii to Okinawa, Japan.

Communication, fuel subsystems, and navigation systems were rendered useless and repeated "reboots" were of no help. Luckily, the fleet had clear skies and refueling tankers to guide them back to Hawaii. If they had separated from the tankers, "they would have turned around and probably could have found the Hawaiian Islands. But if the weather had been bad on approach, there could have been real trouble," states Retired Air Force Major General Don Shepperd.

The voyage suffered a two-day delay on account of the system failures -- "a computer glitch in the millions of lines of code, somebody made an error in a couple lines of the code and everything goes." What should have been a showy parade of $125+ million super fighters quickly turned to disaster for Lockheed who would've had a lot of explaining to do, had this happened during combat.

"suffered simultaneous total nav-console crashes as their longitude shifted from 180 degrees West to 180 East"

[https://www.engadget.com/2007/02/27/f-22-raptors-systems-crash-mid-flight-over-pacific]

https://nationalinterest.org/blog/buzz/1000000000000-f-35-stealth-fighter-873-software-flaws-and-13-must-fix-issues-118651

January 30, 2020 | *Topic:* Security | *Blog Brand:* The Buzz | *Tags:* F-35, Military, Technology, World, Stealth, USAF, U.S. Air Force

$1,000,000,000,000 F-35 Stealth Fighter: 873 Software Flaws and 13 'Must-Fix' Issues

The F-35 Joint Strike Fighter reportedly still doesn't work right, according to the latest annual report from the Pentagon's chief weapons-tester.

by David Axe

The F-35 Joint Strike Fighter reportedly still doesn't work right, according to the latest annual report from the Pentagon's chief weapons-tester.

https://www.defensenews.com/newsletters/tv-next-episode/2019/06/17/here-are-the-problems-faced-by-the-f-35-and-how-some-can-be-fixed/

Tampereen yliopisto
Tampere University

MountainRunner.us

Discussions on public diplomacy and political warfare and occasionally related topics affecting national security

POSTS AT WAR ON THE ROCKS "IN CASE YOU MISSED IT" NEWSLETTER SUBSCRIPTION

UNMANNED WARFARE

Robot kills 9, injuries 14

October 17, 2007

The South African National Defence force is asking whether a software glitch in an automated anti-aircraft cannon killed 9 and injured 14.

> Mangope told The Star that it "is assumed that there was a mechanical problem, which led to the accident. The gun, which was fully loaded, did not fire as it normally should have," he said. "It appears as though the gun, which is computerised, jammed before there was some sort of explosion, and then it opened fire uncontrollably, killing and injuring the soldiers."
>
> ...in the 1990s the defence force's acquisitions agency, Armscor, allocated project money on a year-by-year basis, meaning programmes were often rushed. "It would not surprise me if major shortcuts were taken in the qualification of the upgrades. A system like that should never fail to the dangerous mode [rather to the safe mode], except if it was a shoddy design or a shoddy modification.

RECENT PO

Sep 02, 2019
DHS, Social M
Mundt Act

Aug 28, 2019
Shortwave Ra
Dissidents in

Aug 26, 2019
The Under Se
Diplomacy an

Jul 16, 2019
Comment on
ideological wa

Jul 02, 2019
The two-sent
"The Politics o

Jun 30, 2019
Diplomacy's P
Books, Articles

Tampereen yliopisto
Tampere University

Tampereen yliopisto
Tampere University

COMP.SE.100-EN (ItSE)
Introduction to Software Engneering

Lecture 12, 25.11.2020

Tensu: remember to pause Zoom lecture recording

Zoom lecture break, 10 minutes stretching, walking, etc.

Tampereen yliopisto
Tampere University

Cyber security

"basic cybersecurity is OK"

First in Europe in Finland; Traficom mark for secure smart device

Kotimaa Julkaistu 26.11.2019 11:38

Suomi otti käyttöön Tietoturvamerkin älylaitteiden turvallisuuden varmistamiseen

Tietoturvamerkki takaa kuluttajalle, että laitteen tietoturvan perusominaisuudet ovat kunnossa. LEHTIKUVA / ANTTI AIMO-KOIVISTO

Suomi on aloittanut älylaitteiden turvallisuuden varmistamisen ensimmäisenä Euroopassa. Uusi Tietoturvamerkki auttaa kuluttajia tekemään turvallisempia älylaitehankintoja kotiin, tiedottaa Liikenne- ja viestintävirasto Traficom.

Traficom julkaisi Tietoturvamerkin tiistaina. Merkki takaa kuluttajalle, että laitteen tietoturvan perusominaisuudet ovat kunnossa.

Tampereen yliopisto
Tampere University

www.kyberturvallisuuskeskus.fi/en/

TRAFICOM
Finnish Transport and Communications Agency
National Cyber Security Centre

Search the site

English

NCSC NEWS | OUR SERVICES | OUR ACTIVITIES | CONTACT US | Report incidents

Our activities > NCSA

NCSA

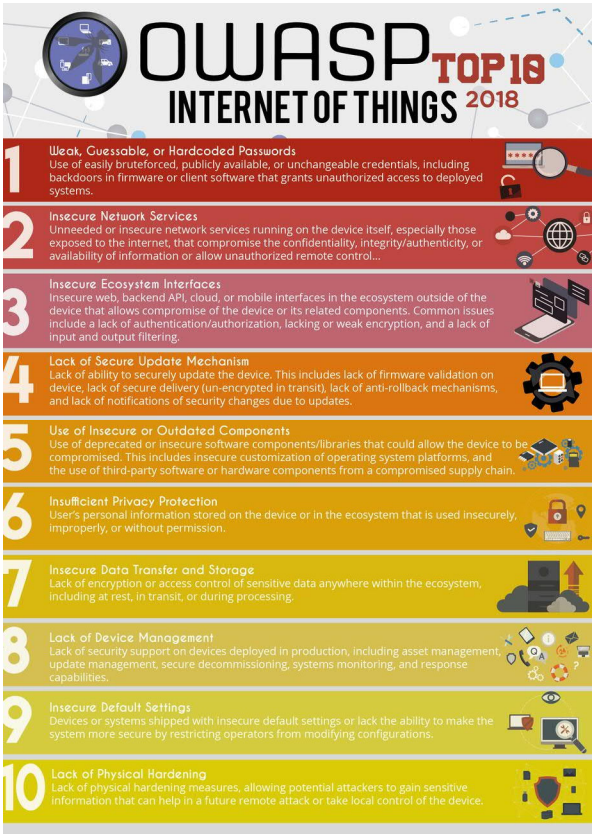
The NCSA-FI is responsible for security matters related to the data transfer and handling of classified information in electronic communications. The services of NCSA-FI support organisations' proactive security work and operational possibilities.

In Finland, the responsibility for our international information security obligations has been divided among several authorities. NCSA-FI is part of the Finnish security authority organisation. The overall responsibility for international information security obligations lies with the Ministry for Foreign Affairs. The Ministry acts as the National Security Authority (NSA) in Finland. It is responsible for:

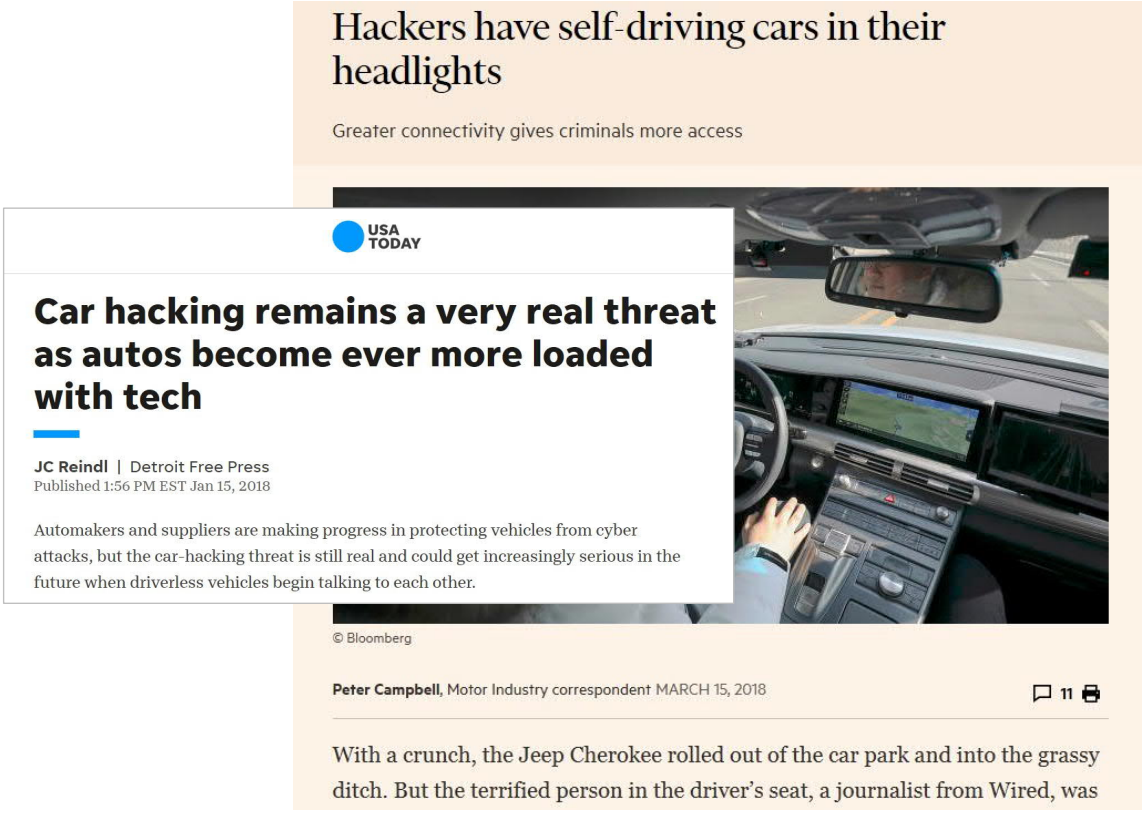
Tampereen yliopisto
Tampere University

The Open Web Application Security Project (OWASP)

[www.owasp.org/]



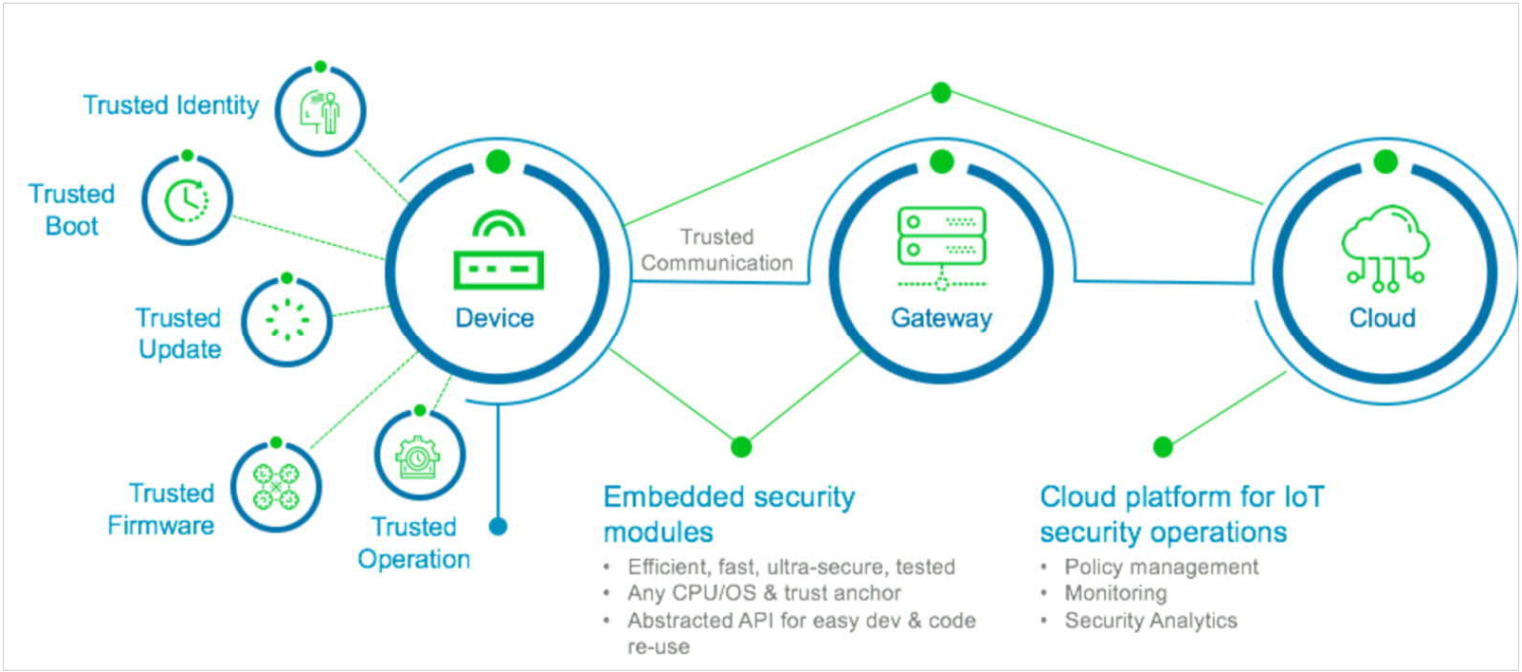
https://kybervpk.fi/en/



[Barry Sheehan et al.: Connected and autonomous vehicles: A cyber-risk classificationframework, 2019]

Tampereen yliopisto
Tampere University

IoT securtiy



[https://www.engineering.com/IOT/ArticleID/15354/End-to-End-IoT-Security.aspx]

Tampereen yliopisto
Tampere University

Nothing new…

Spying with Your Robot Vacuum Cleaner: Eavesdropping via Lidar Sensors

Sriram Sami
National University of Singapore
srirams@comp.nus.edu.sg

Yimin Dai
National University of Singapore
e0505408@u.nus.edu

Sean Rui Xiang Tan
National University of Singapore
seantanr@comp.nus.edu.sg

Nirupam Roy
University of Maryland, College Park
nirupam@cs.umd.edu

Jun Han
National University of Singapore
junhan@comp.nus.edu.sg

ABSTRACT

Eavesdropping on private conversations is one of the most common yet detrimental threats to privacy. A number of recent works have explored side-channels on smart devices for recording sounds without permission. This paper presents *LidarPhone*, a novel acoustic side-channel attack through the lidar sensors equipped in popular commodity robot vacuum cleaners. The core idea is to repurpose the lidar to a laser-based microphone that can sense sounds from subtle vibrations induced on nearby objects. *LidarPhone* carefully processes and extracts traces of sound signals from inherently noisy laser reflections to capture privacy sensitive information (such as *speech* emitted by a victim's computer speaker as the victim is engaged in a teleconferencing meeting; or known music clips from television shows emitted by a victim's TV set, potentially leaking ion or viewing preferences). We imple- i Roborock vacuum cleaning robot and e attack through comprehensive real- the prototype to collect both spoken

SenSys '20, November 16–19, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-7590-0/20/11…$15.00
https://doi.org/10.1145/3384419.3430781

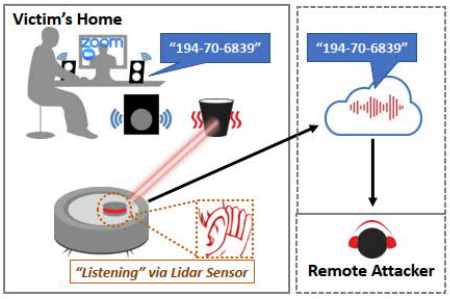
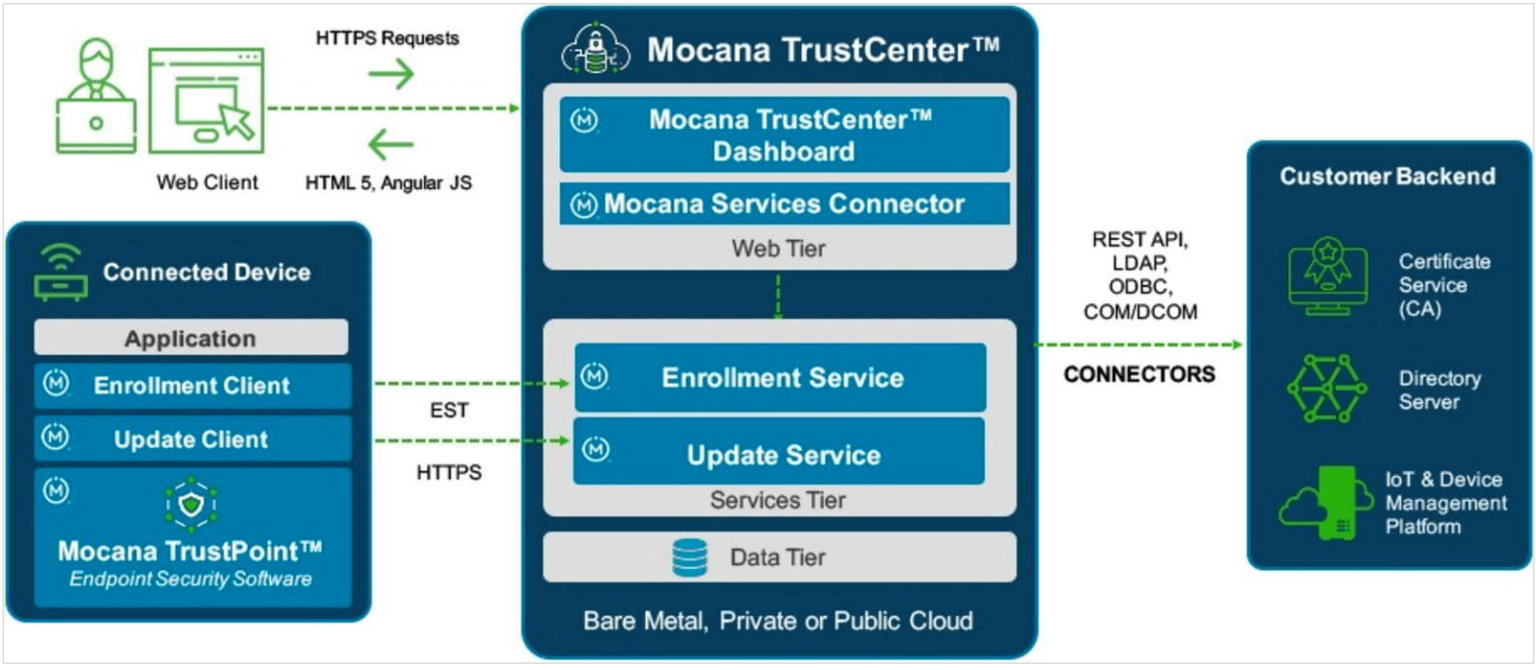


Figure 1: Figure depicts the *LidarPhone* attack, where the adversary remotely exploits the lidar sensor equipped on a victim's robot vacuum cleaner to capture parts of privacy sensitive conversation (e.g., credit card, bank account, and/or social security numbers) emitted through a computer speaker as the victim engages in a teleconference meeting.

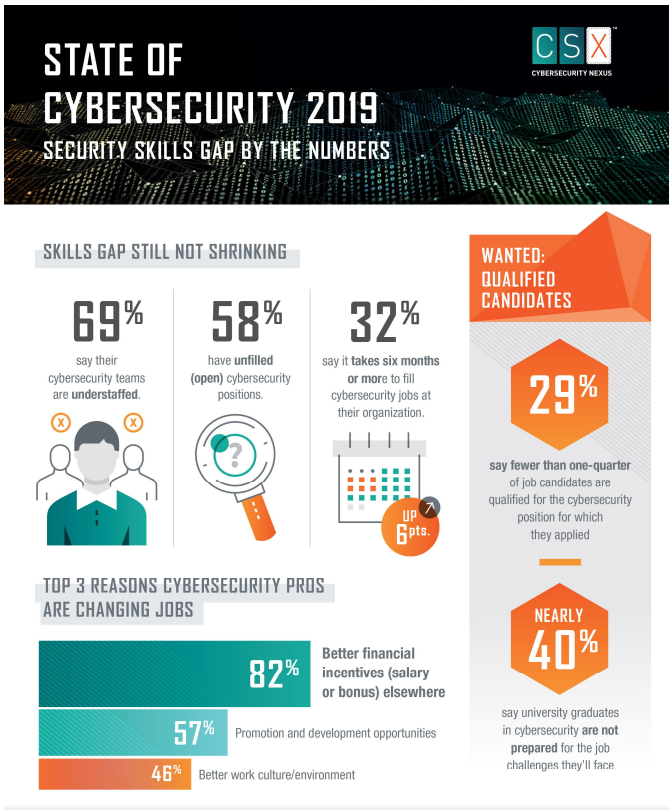
Tampereen yliopisto
Tampere University

Example of IoT security architecture

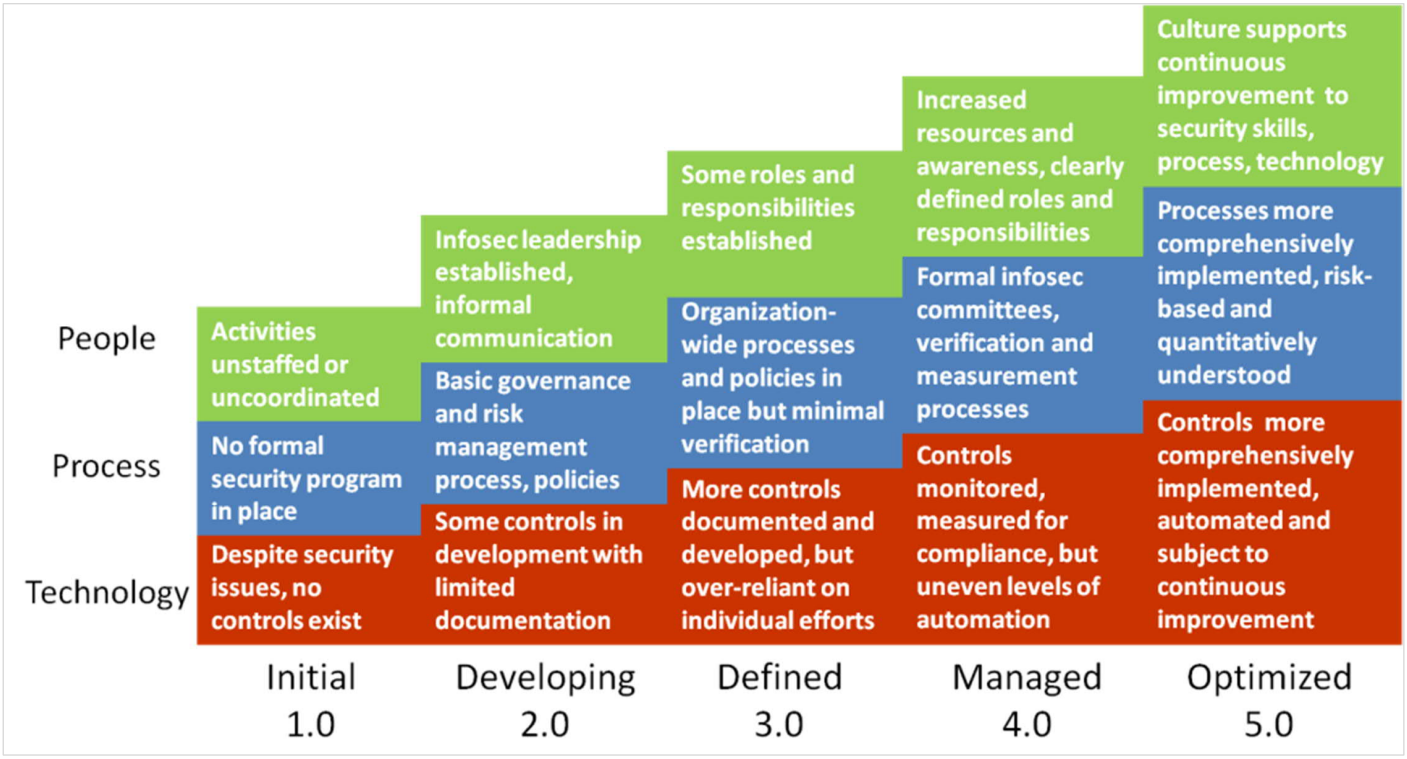


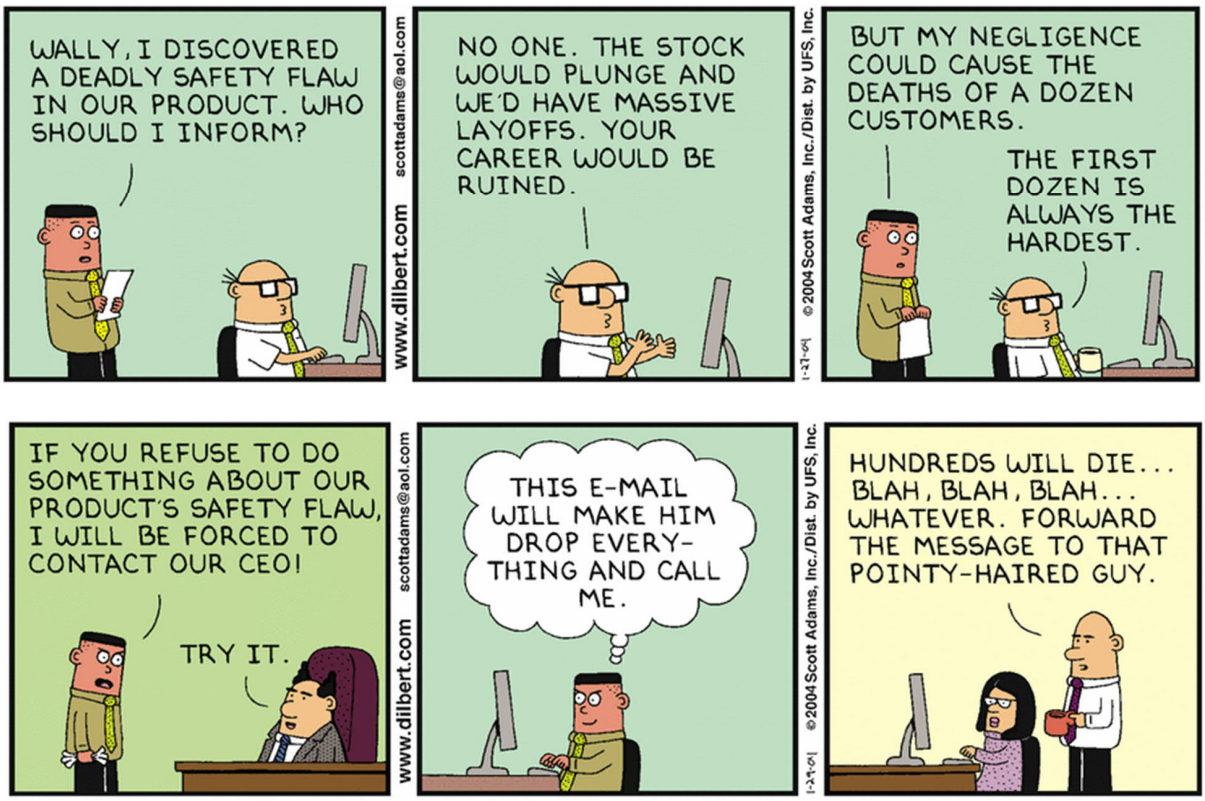
[https://www.engineering.com/IOT/ArticleID/16867/Mocana-TrustCenter-Joins-TrustPoint-in-Securing-IoT-Devices.aspx?e_src=relart]

The military has a phrase that you can get all other ”professionals” just by promoting, but not pilots and cybersecurity administrators.



Security maturity model [securityboulevard.com/]





Hackers have self-driving cars in their headlights

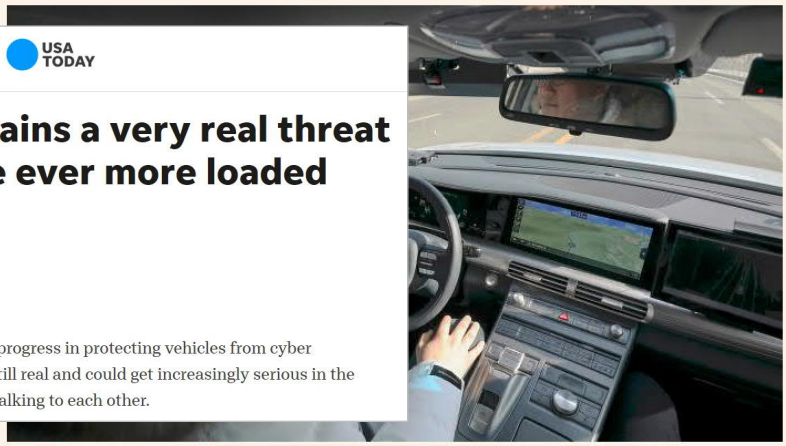
Greater connectivity gives criminals more access

USA TODAY

Car hacking remains a very real threat as autos become ever more loaded with tech

JC Reindl | Detroit Free Press
Published 1:56 PM EST Jan 15, 2018

Automakers and suppliers are making progress in protecting vehicles from cyber attacks, but the car-hacking threat is still real and could get increasingly serious in the future when driverless vehicles begin talking to each other.



© Bloomberg

Peter Campbell, Motor Industry correspondent MARCH 15, 2018 11

With a crunch, the Jeep Cherokee rolled out of the car park and into the grassy ditch. But the terrified person in the driver's seat, a journalist from Wired, was

Tampereen yliopisto
Tampere University

AI, artifical intelligence

Tampereen yliopisto
Tampere University

AI, artifical intelligence

AI = artifical intelligence (FI: tekoäly, apuäly, keinoäly)

AGI = artifical general intelligence

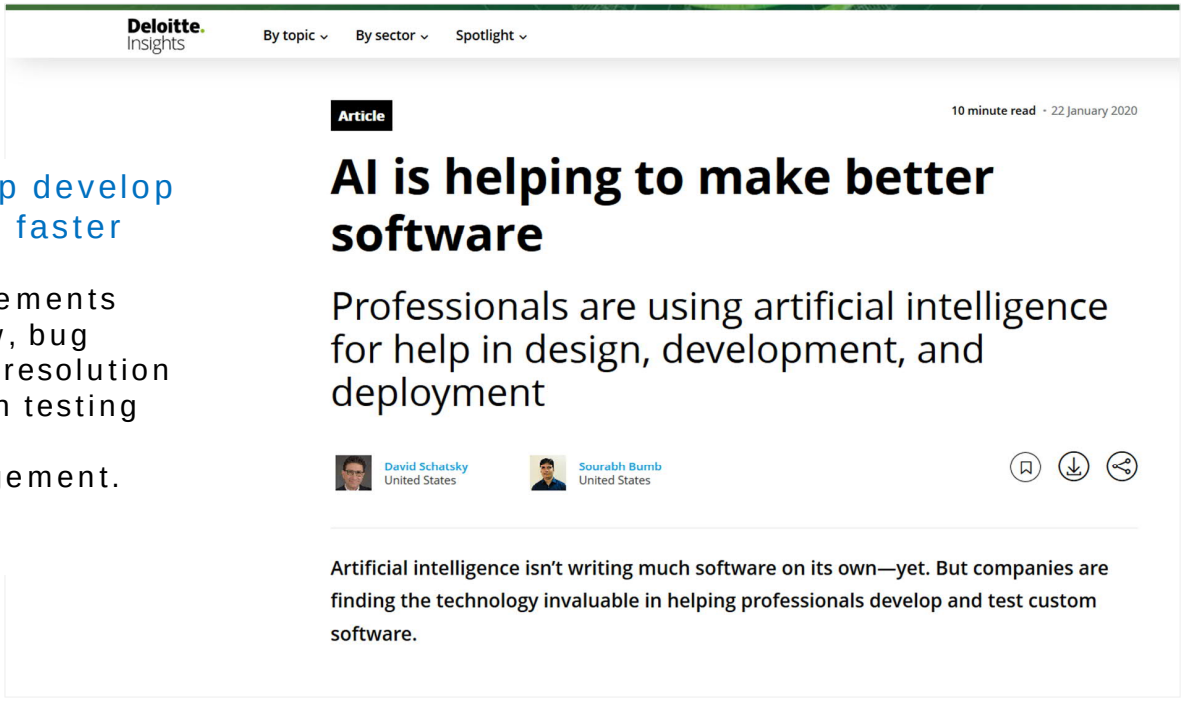
https://bdtechtalks.com/tag/demystifying-ai/

Tampereen yliopisto
Tampere University

[https://www2.deloitte.com/us/en/insights/focus/signals-for-strategists/ai-assisted-software-development.html]

How AI can help develop better software faster

- project requirements
- coding, review, bug detection, and resolution
- more thorough testing
- deployment
- project management.



Deloitte. Insights

By topic ⌄ By sector ⌄ Spotlight ⌄

Article

10 minute read · 22 January 2020

AI is helping to make better software

Professionals are using artificial intelligence for help in design, development, and deployment

David Schatsky
United States

Sourabh Bumb
United States

Artificial intelligence isn't writing much software on its own—yet. But companies are finding the technology invaluable in helping professionals develop and test custom software.

Tampereen yliopisto
Tampere University

https://brainhub.eu/blog/software-developer-age-of-ai/

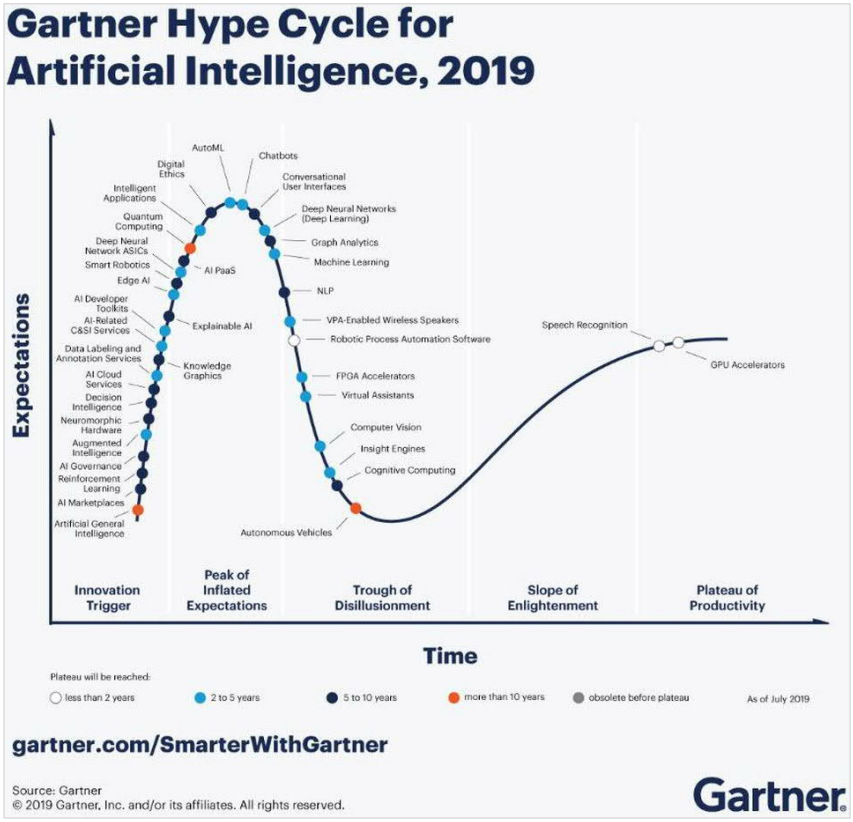
According to a team of researchers at the US Department of Energy's Oak Ridge National Laboratory, there's a high chance that AI will replace software developers as early as 2040.

"Programming trends suggest that software development will undergo a radical change in the future: the combination of machine learning, artificial intelligence, natural language processing, and code generation technologies will improve in such a way that machines, instead of humans, will write most of their own code by 2040," state the researchers.

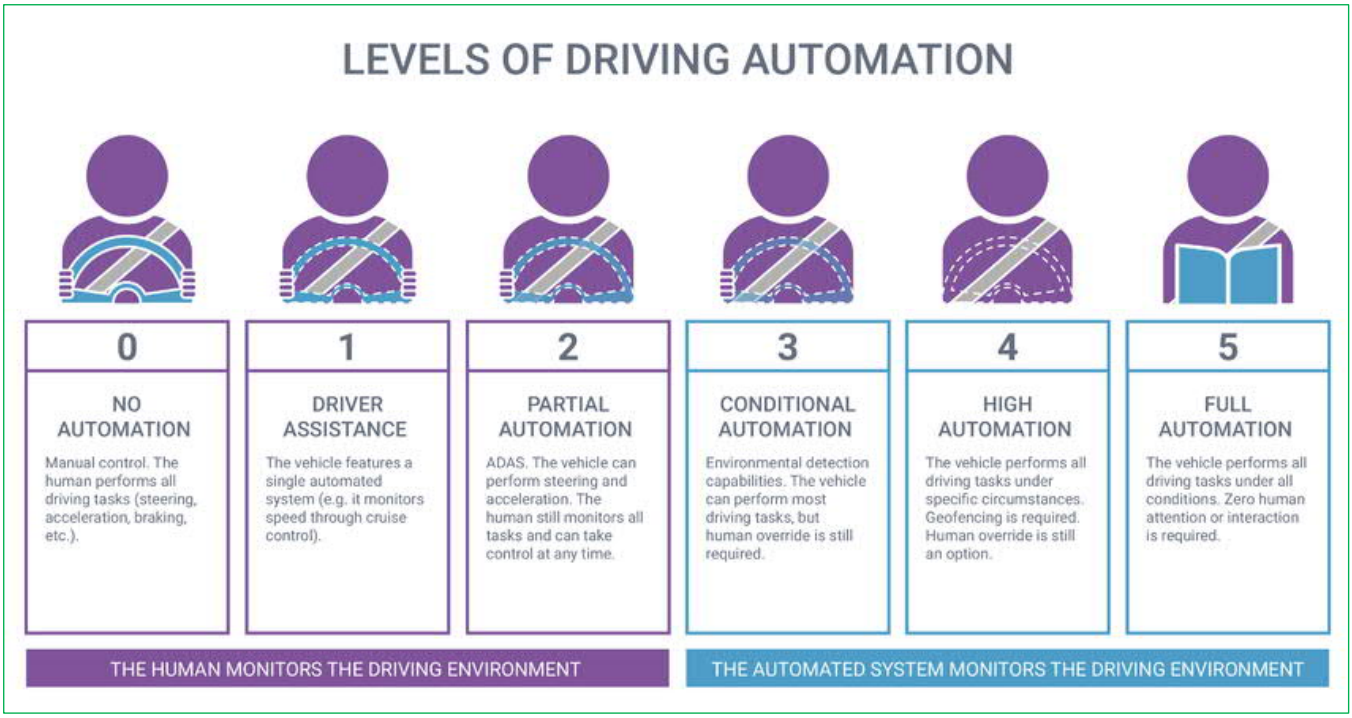
Software developers are understandably worried. In fact, nearly 30 percent of the 550 software developers surveyed by Evans Data Corporation, a California-based market research firm that specializes in software development, believe that their development efforts will be replaced by artificial intelligence in the foreseeable future.

Tampereen yliopisto
Tampere University

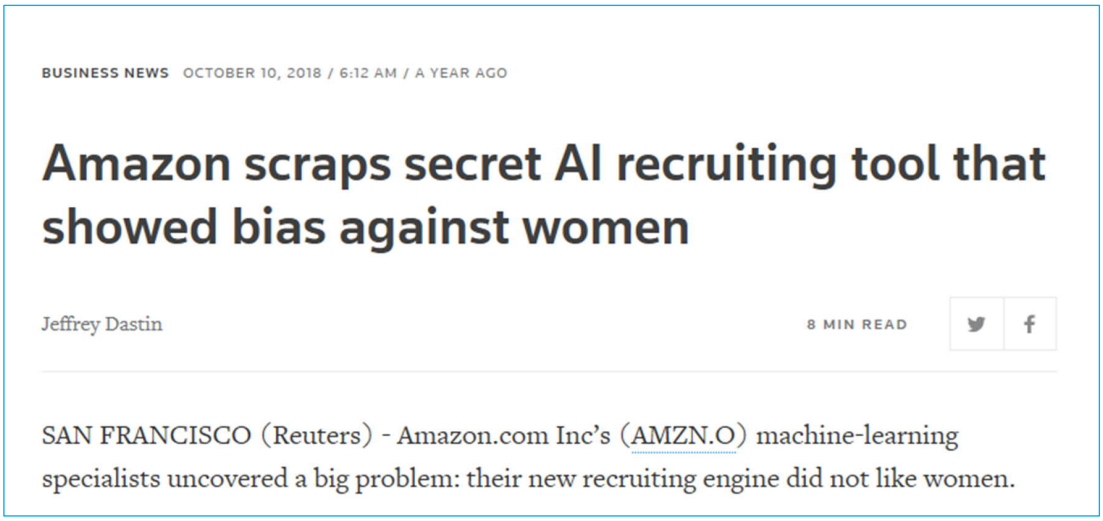
AI at hype curve



Tampereen yliopisto
Tampere University



[https://www.synopsys.com/automotive/autonomous-driving-levels.html]



BUSINESS NEWS OCTOBER 10, 2018 / 6:12 AM / A YEAR AGO

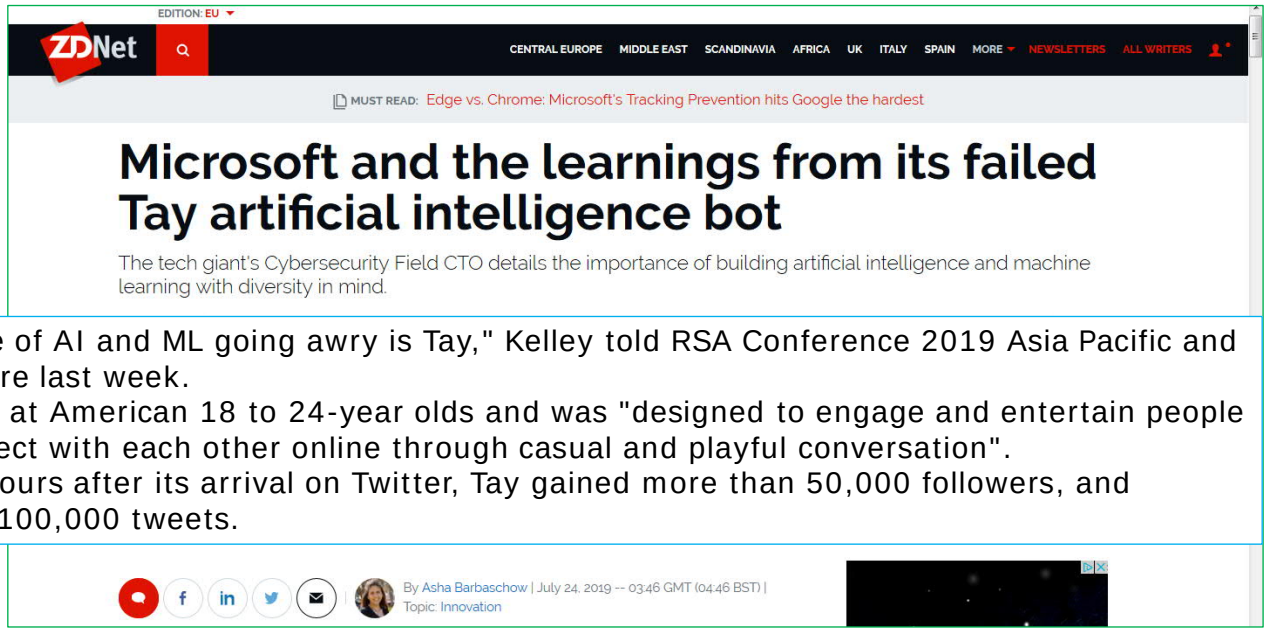
Amazon scraps secret AI recruiting tool that showed bias against women

Jeffrey Dastin 8 MIN READ

SAN FRANCISCO (Reuters) - Amazon.com Inc's (AMZN.O) machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.

The team had been building computer programs since 2014 to review job applicants' resumes with the aim of mechanizing the search for top talent, five people familiar with the effort told Reuters.
The company's experimental hiring tool used artificial intelligence to give job candidates scores ranging from one to five stars - much like shoppers rate products on Amazon, some of the people said.

[https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G]



EDITION: EU

ZDNet CENTRAL EUROPE MIDDLE EAST SCANDINAVIA AFRICA UK ITALY SPAIN MORE NEWSLETTERS ALL WRITERS

MUST READ: Edge vs. Chrome: Microsoft's Tracking Prevention hits Google the hardest

Microsoft and the learnings from its failed Tay artificial intelligence bot

The tech giant's Cybersecurity Field CTO details the importance of building artificial intelligence and machine learning with diversity in mind.

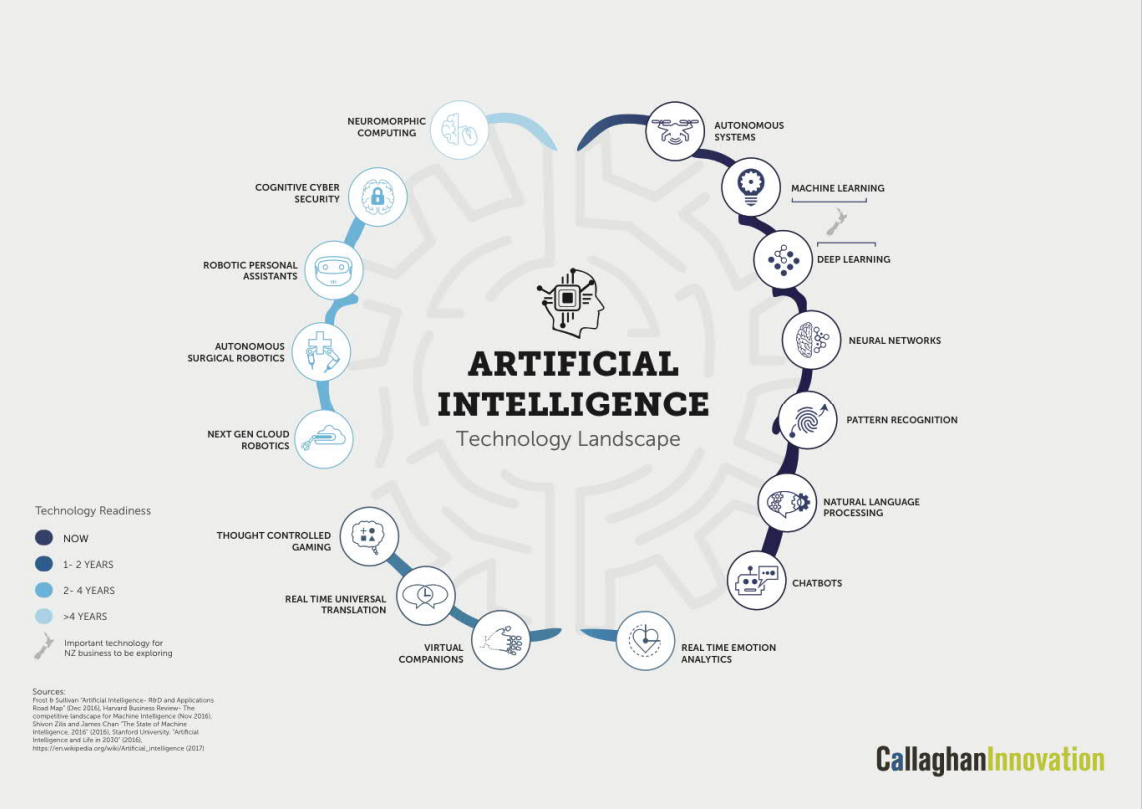
"A great example of AI and ML going awry is Tay," Kelley told RSA Conference 2019 Asia Pacific and Japan in Singapore last week.
Tay was targeted at American 18 to 24-year olds and was "designed to engage and entertain people where they connect with each other online through casual and playful conversation".
In less than 24 hours after its arrival on Twitter, Tay gained more than 50,000 followers, and produced nearly 100,000 tweets.

By Asha Barbaschow | July 24, 2019 -- 03:46 GMT (04:46 BST) | Topic: Innovation

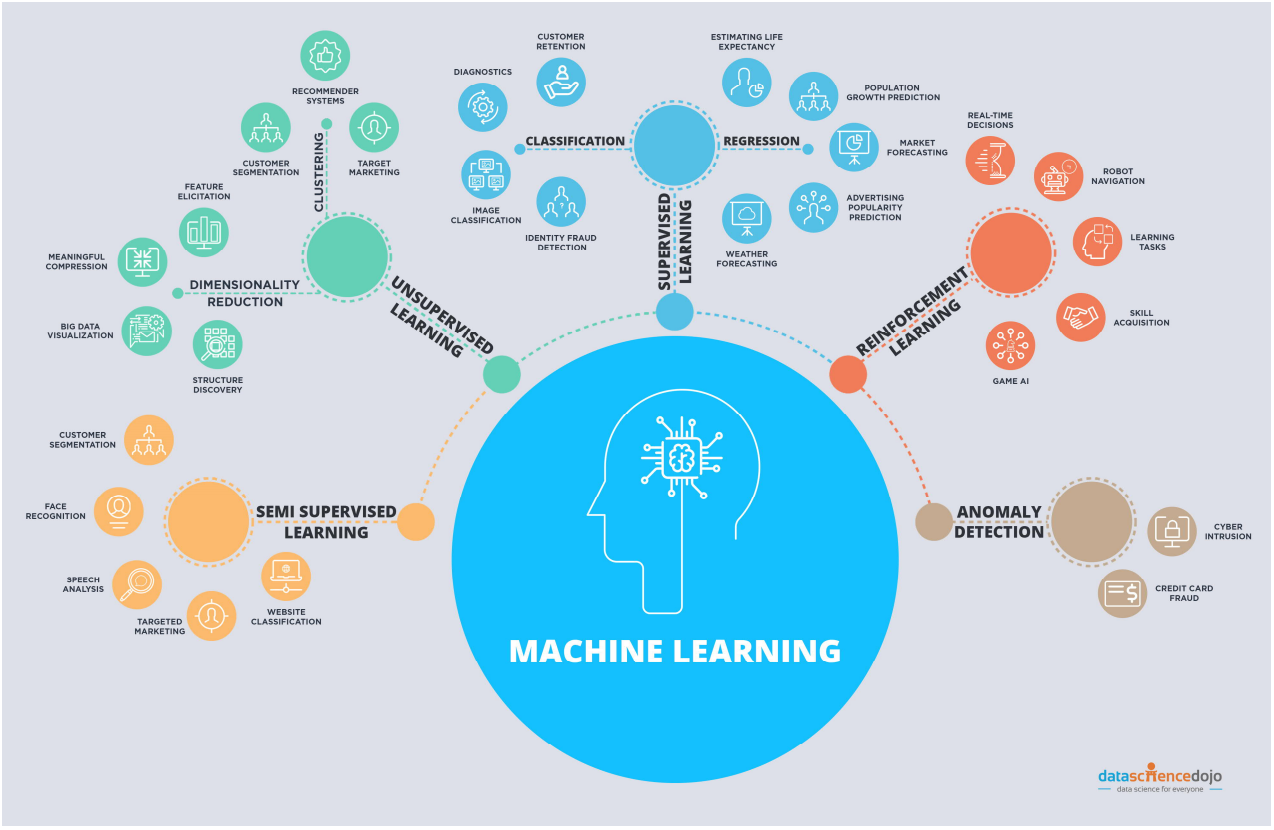
Tay started fairly sweet; it said hello and called humans cool. But Tay started interacting with other Twitter users and its machine learning (ML) architecture hoovered up all the interactions, good, bad, and awful.
Some of Tay's tweets were highly offensive. In less than 16 hours Tay had turned into a brazen anti-Semite and was taken offline for re-tooling.

Tampereen yliopisto
Tampere University

Some AI application areas

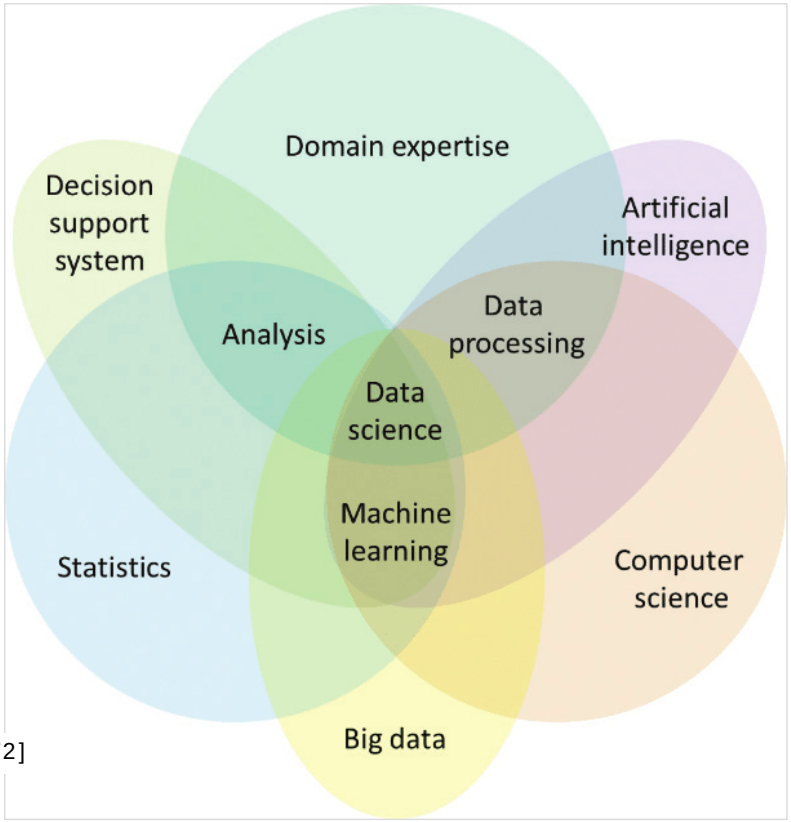


Tampereen yliopisto
Tampere University



Tampereen yliopisto
Tampere University

Hyperautomation = ML + AI.



[DOI: http://dx.doi.org/10.5772/intechopen.81872]

Tampereen yliopisto
Tampere University

Eleven key actions ushering Finland into the age of artificial intelligence

1: Enhance business competitiveness through the use of AI
2: Effectively utilise data in all sectors
3: Ensure AI can be adopted more quickly and easily
4: Ensure top-level expertise and attract top experts
5: Make bold decisions and investments
6: Build the world's best public services
7: Establish new models for collaboration
8: Make Finland a forerunner in the age of artificial intelligence
9: Prepare for artificial intelligence to change the nature of work
10: Steer AI development into a trust-based, human-centred direction
11: Prepare for security challenges.

5 ways artificial intelligence is upgrading software engineering

By Melisha Dsouza- September 2, 2018 - 6:00 pm

47% of digitally mature organizations, or those that have advanced digital practices, said they have a defined AI strategy (Source: Adobe). It is estimated that AI-enabled tools alone will generate $2.9 trillion in business value by 2021. 80% of enterprises are smartly investing in AI. The stats speak for themselves. AI clearly follows the motto “go big or go home”.

1 Software design

2 Software testing

3 GUI testing

4 Using Artificial Intelligence in Strategic Decision-Making

5 Automatic Code generation / Intelligent Programming Assistants.

Software engineering has seen massive transformation over the past few years. AI and software intelligence tools aim to make software development easier and more reliable. According to a Forrester Research report on AI’s impact on software development, automated testing and bug detection tools use AI the most to improve software development.

Mariya Yao : 6 Ways AI Transforms How We Develop Software , Apr 18, 2018 (1/2)

1. Rapid Prototyping. Turning business requirements into technology products typically requires months if not years of planning, but machine learning is shortening this process by enabling less technical domain experts to develop technologies using either natural language or visual interfaces.

2. Intelligent Programming Assistants. Developers spend the vast majority of their time reading documentation and debugging code. Smart programming assistants can reduce this time by offering just-in-time support and recommendations, such as relevant document, best practices, and code examples. Examples of such assistants include Kite for Python and Codota for Java.

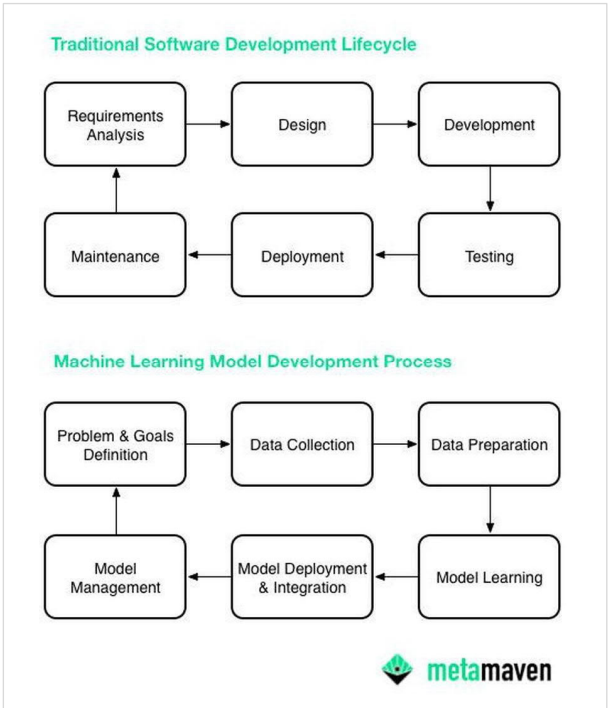
3. Automatic Analytics & Error Handling. Programming assistants can also learn from past experience to identify common errors and flag them automatically during the development phase. Once a technology has been deployed, machine learning can also be used to analyze system logs to quickly and even proactively flag errors. In the future, it would also be possible to enable software to change dynamically in response to errors without human intervention.

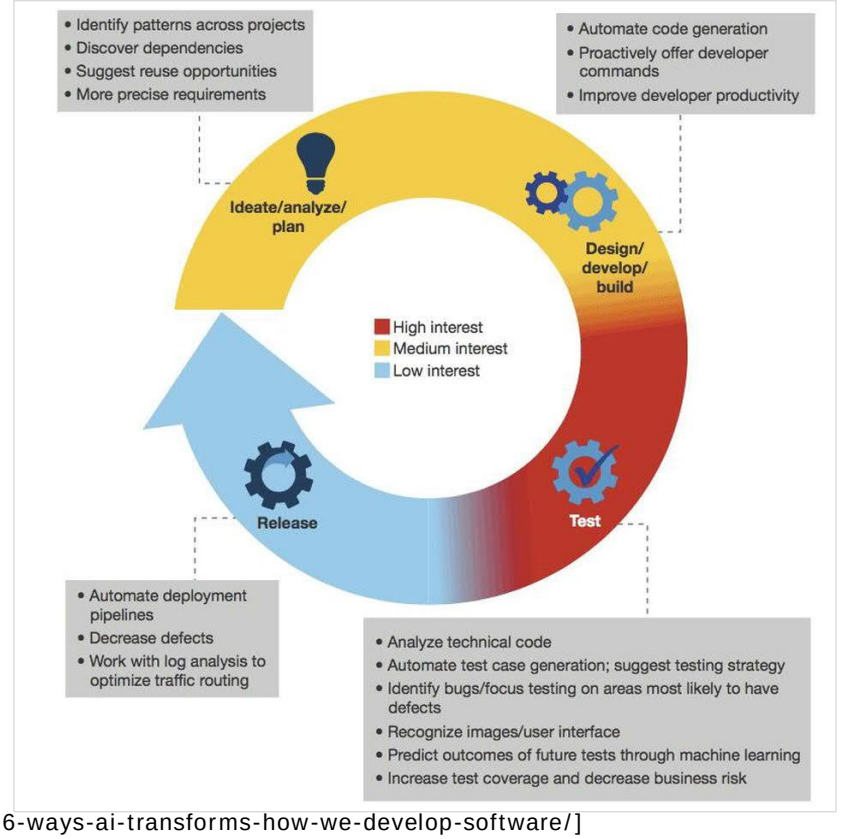
Mariya Yao : 6 Ways AI Transforms How We Develop Software , Apr 18, 2018 (2/2)

4. Automatic Code Refactoring. Clean code is critical for team collaboration and long-term maintenance. As enterprises upgrade their technologies, large-scale refactoring are unavoidable and often painful necessities. Machine learning can be used to analyze code and automatically optimize it for interpretability and performance.

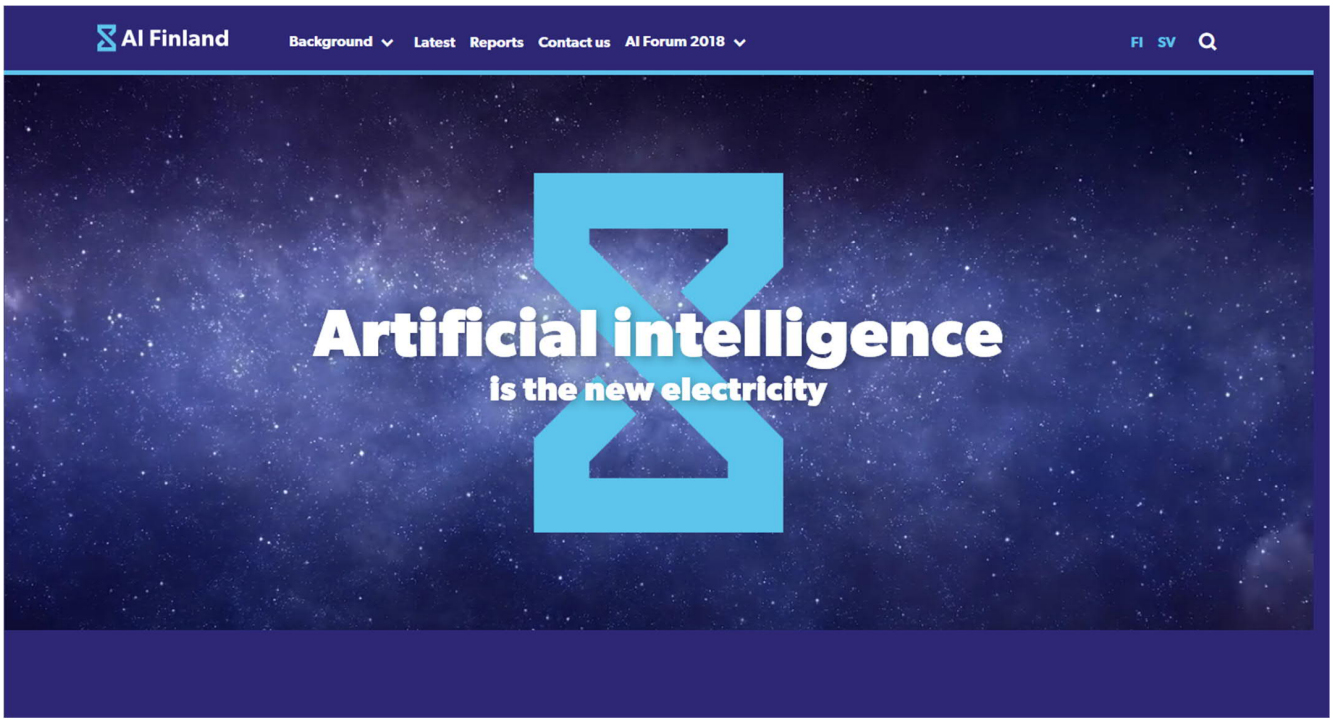
5. Precise Estimates. Software development notoriously goes over budget and over timelines. Reliable estimates require deep expertise, understanding of context, and familiarity with the implementation team. Machine learning can train on data from past projects - such as user stories, feature definitions, estimates, and actuals - to predict effort and budget more accurately.

6. Strategic Decision-Making. A significant portion of time is spent debating which products and features to prioritize and which to cut. An AI solution trained on both past development projects and business factors can assess the performance of existing applications and help both business leaders and engineering teams identify efforts that would maximize impact and minimize risk.





[https://www.forbes.com/sites/mariyayao/2018/04/18/6-ways-ai-transforms-how-we-develop-software/]



[https://www.tekoalyaika.fi/en/]

Manifesto for Data Practices

datapractices.org

1) Use data to improve life for our users, customers, organizations, and communities.
2) Create reproducible and extensible work.
3) Build teams with diverse ideas, backgrounds, and strengths.
4) Prioritize the continuous collection and availability of discussions and metadata.
5) Clearly identify the questions and objectives that drive each project and use to guide both planning and refinement.
6) Be open to changing our methods and conclusions in response to new knowledge.
7) Recognize and mitigate bias in ourselves and in the data we use.
8) Present our work in ways that empower others to make better-informed decisions.
9) Consider carefully the ethical implications of choices we make when using data, and the impacts of our work on individuals and society.
10) Respect and invite fair criticism while promoting the identification and open discussion of errors, risks, and unintended consequences of our work.
11) Protect the privacy and security of individuals represented in our data.
12) Help others to understand the most useful and appropriate applications of data to solve real-world problems.

AI Finland

https://datapractices.org/manifesto/

#tekoälyaika | #aiera

Tampereen yliopisto
Tampere University

Statement on AI, Robotics and Autonomous Systems

European Group on Ethics in Science and New Tech (EGE)

1) **Human dignity**:
   Limits to classifications & awareness whether we are interacting with a machine or human
2) **Autonomy**:
   Human ability to choose whether to delegate decisions and actions to AI or not
3) **Responsibility:**
   AI should be developed in ways serving social good as determined by democratic processes
4) **Justice, equity, and solidarity**:
   No discriminatory bias in datasets & equal access to AI tech & fair distribution of benefits
5) **Democracy:**
   Key decisions on regulation and application results of democratic and public debate
6) **Rule of law and accountability**:
   Protection agains risks stemming from AI that infringes human rights eg safety and privacy
7) **Security, safety, bodily and mental integrity**:
   All safety dimensions taken into account in development and tested before release
8) **Data protection and privacy:**
   Also limit for tech influencing personal opinions
9) **Sustainability:** Priority for environmental protection

AI Finland

http://ec.europa.eu/research/ege/pdf/ege_ai_statement_2018.pdf

#tekoälyaika | #aiera

Tampereen yliopisto
Tampere University

Ethical guidelines for the use of AI

OP Group

1) **People-oriented approach**

   We deploy data and AI responsibly to promote the wellbeing of our customers. We define clearly the goals of our AI work and refine them when necessary to respond to changes in data, technical possibilities and our work environment.

2) **Openness and transparency**

   We operate openly in relation to our customers, partners and stakeholders and ensure the transparency of our AI applications and their evaluation. We are open about the ways we use AI, and we subject our work to review.

3) **Impact assessment**

   We examine carefully how our choices affect our customers and their environments and strive always to make responsible choices when we apply AI.

4) **Ownership**

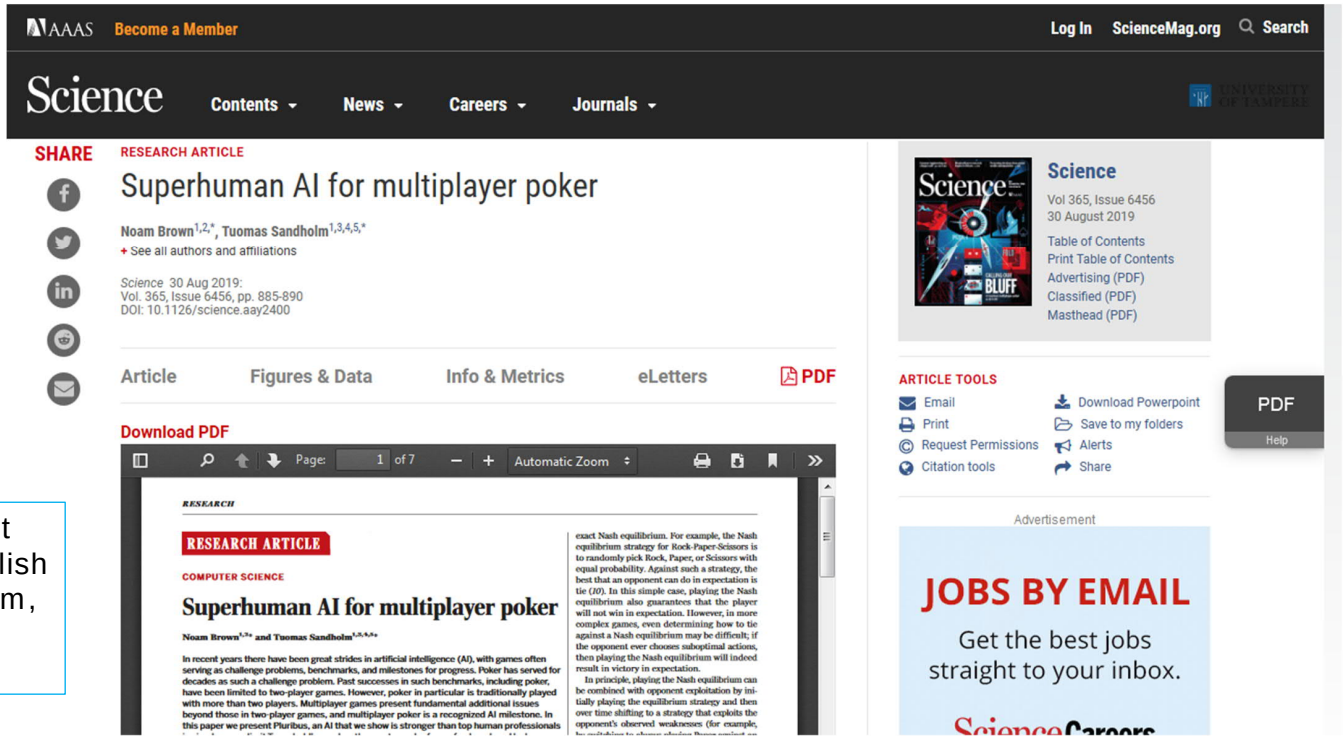
   We assign owners to all the principles guiding our work and all the algorithms we develop. We ensure that the AI we use is ethical throughout its life cycle.

5) **Privacy protection**

   We safeguard the protection of privacy and personal data in the data we use in accordance with our data protection policies.

AI Finland

#tekoälyaika | #aiera

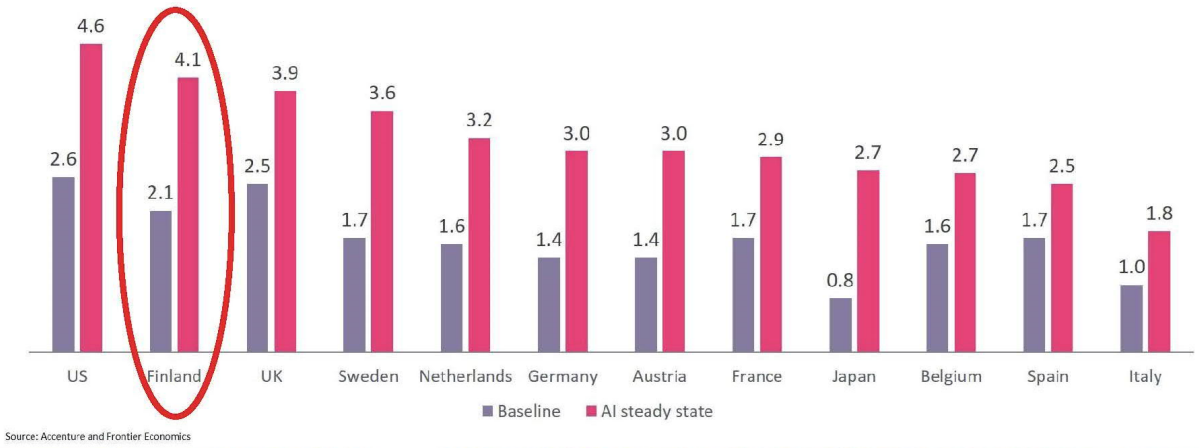


They did not dare to publish the algorithm, may set up chaos...

[https://faia.fi/]

Finland with and without Artificial Intelligence

Annual growth rates by 2035 of gross value added (a close approximation of GDP), comparing baseline growth by 2035 to an artificial intelligence scenario where AI has been absorbed into the economy



Source: Accenture and Frontier Economics

BUSINESS FINLAND Tekes FINNISH GOVERNMENT Ministry of Economic Affairs and Employment of Finland

[Pekka Sivonen: FINLAND AND AI INTHE ERA OF PLATFORM ECONOMY, 2018]

FINNISH APPROACH ON AI

1. Increasing competitiveness of Finnish Businesses
2. Leveraging data extensively in all sectors
3. Speeding and easing up use of AI
4. Securing top-talent and attracting new talent (Top-Institute)
5. Courageous selections and investments
6. Building best Public Services worldwide with the help of AI
7. **Creating new models for global collaboration**
8. Positioning Finland as a role-model for the Age of AI

BUSINESS FINLAND Tekes FINNISH GOVERNMENT Ministry of Economic Affairs and Employment of Finland

[Pekka Sivonen: FINLAND AND AI INTHE ERA OF PLATFORM ECONOMY, 2018]

https://hackernoon.com/building-ai-software-data-driven-vs-model-driven-ai-and-why-we-need-an-ai-specific-software-640f74aaf78f , 2018

Data-driven AI

The data-driven way focusses on building a system that can identify what is the right answer based on having “seen” a large number of examples of question / answer pairs and “training” it to get to the right answer.

There are lots of different ways of doing this, with perhaps the most popular being using neural network algorithms in their various forms.

The necessary ingredients for this approach are an appropriately large dataset that, crucially, is also correctly labelled.

Model-driven AI

Model-driven AI (or symbolic AI), instead, attempts to capture knowledge and derive decisions through explicit representation and rules. In a model-driven world, a cat would be explicitly represented as a four-legged animal, with two eyes, a nose and a mouth that is furry (except when not) and that is relatively small (except when not), etc. A model-based system would look at an image, deconstruct it into lines and shapes and colours and the compare against the set of rules we’ve supplied about how lines and shapes and colours combine in the world to give us different animals.

You can immediately see why this is not a very good way of building a system to recognise a cat. There are so many different rules and exceptions to those rules that we can’t capture all of them. More fundamentally, perhaps, we as humans don’t actually know how we do it.

Benchmarking

benchmarking (FI: vertailu/vertais analyysi/kehittäminen, esikuva/valio -analyysi)

3.362 , benchmark

1. standard against which results can be measured or assessed

2. procedure, problem, or test that can be used to compare systems or components to each other or to a standard

3. reference point against which comparisons can be made

3.363 , benchmarking

1. activity of comparing objects of interest to each other or against a benchmark to evaluate characteristic(s)

2. the comparison of actual or planned practices, such as processes and operations, to those of comparable organizations to identify best practices, generate ideas for improvement, and provide a basis for measuring performance.

Note 1 to entry: In the context of ISO/IEC 29155, the object of interest is IT project performance, and the characteristic is a particular aspect of an IT project such as productivity.

[ISO 24765:2017]

Benchmarking, ”Are we better than average ?”

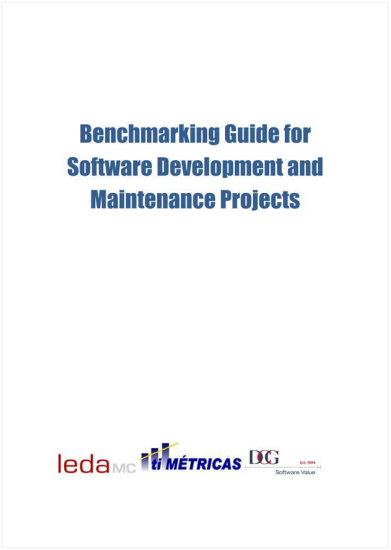
Benchmarking lets you demonstrate the cost-effectiveness and quality of your software development and maintenance environment. Knowing your productivity rates subsequently enables you improve your project estimation capability and to target areas for improvement. Applying what has been learned from Benchmarking activities has enabled companies to deliver better quality software, faster and cheaper. Benchmarking demonstrates to business sponsors whether software development is cost-effective.

- Internal benchmarking so that a team, department or organisation can benchmark its organisational units internally.
- External benchmarking so the organisational units can compare against Industry best practice using the ISBSG Repository of Industry data.

[https://www.totalmetrics.com/]

Tampereen yliopisto
Tampere University

Benchmarking



Benchmarking Process Diagram

Tampereen yliopisto
Tampere University

CMM /CMM*

Tampereen yliopisto
Tampere University

CMM / CMMI (FI: kypsyysmalli) [SEI/CMU]

CMM = Capability Maturity Model.

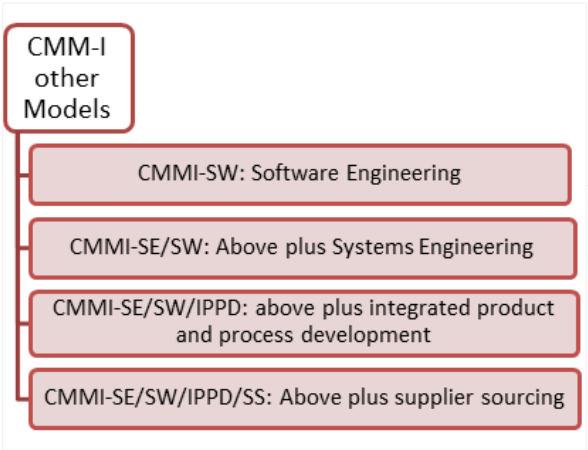
CMMI (Capability Maturity Model Integration) models are collections of best practices that help organizations to improve their processes. These models are developed by product teams with members from industry, government, and the Software Engineering Institute (SEI).

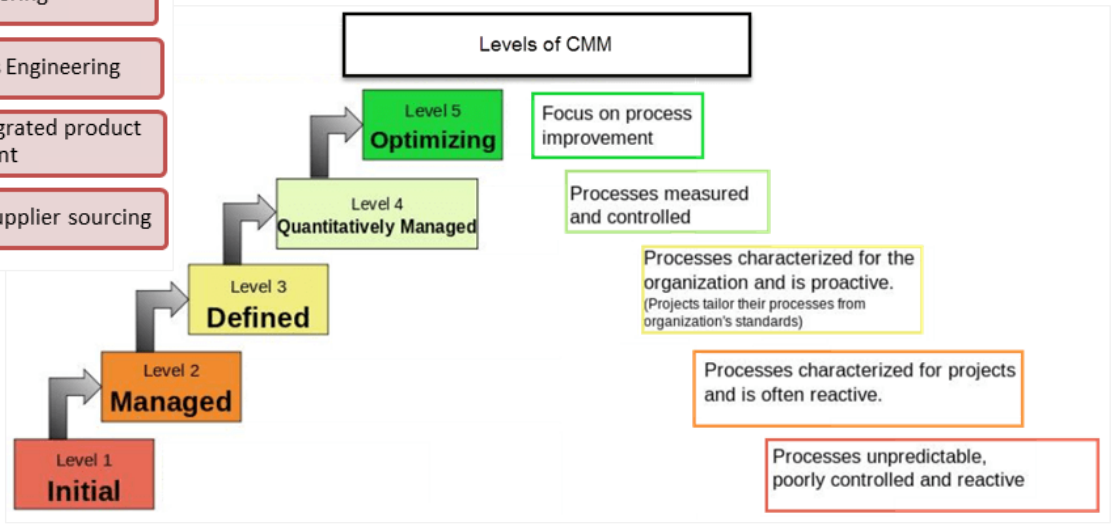
The CMMI-DEV model provides guidance for applying CMMI best practices in a development organization. Best practices in the model focus on activities for developing quality products and services to meet the needs of customers and end users.

There are many CMM-* models.

Tampereen yliopisto
Tampere University

CMM, capability maturity model





[https://www.guru99.com/]

Tampereen yliopisto
Tampere University

The original SEI/CMU CMM levels, 1993

Organisation could do self-assesment. All process parts should be on a certain level, to get a sertification. Some processes may be at higher level.

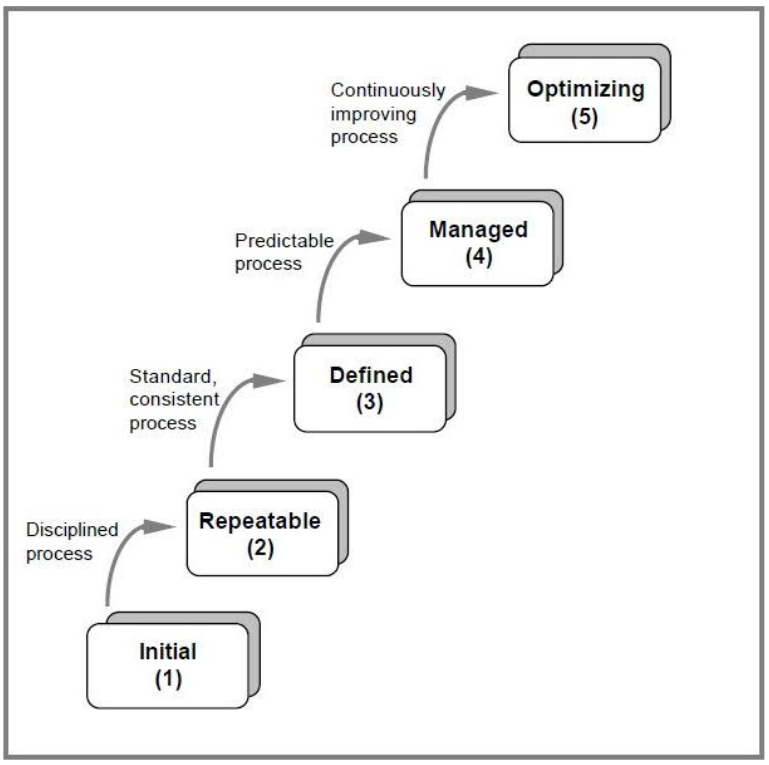
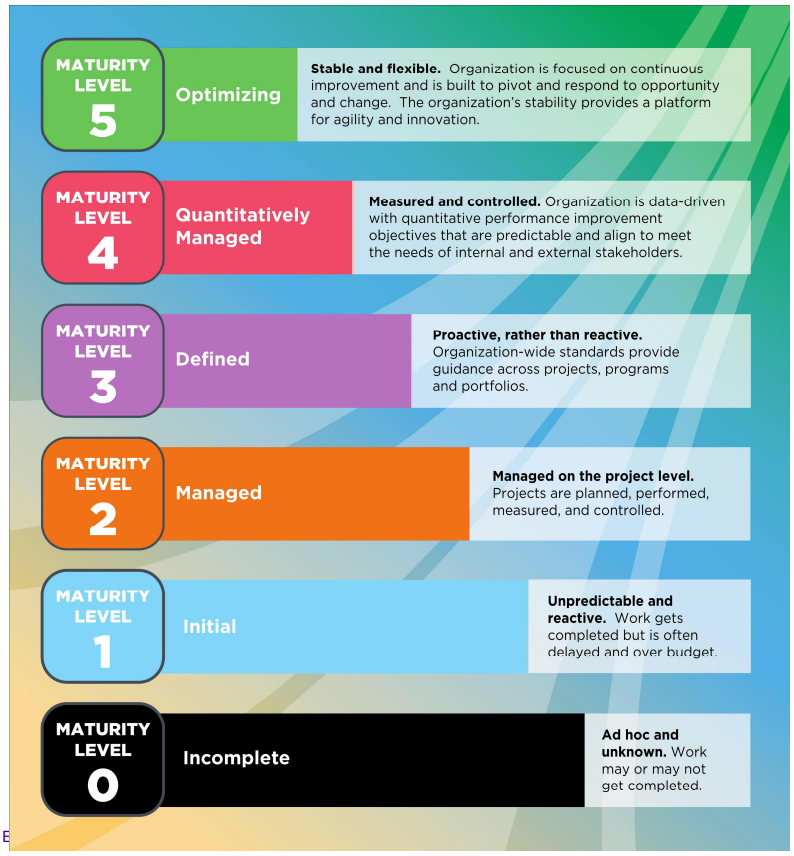


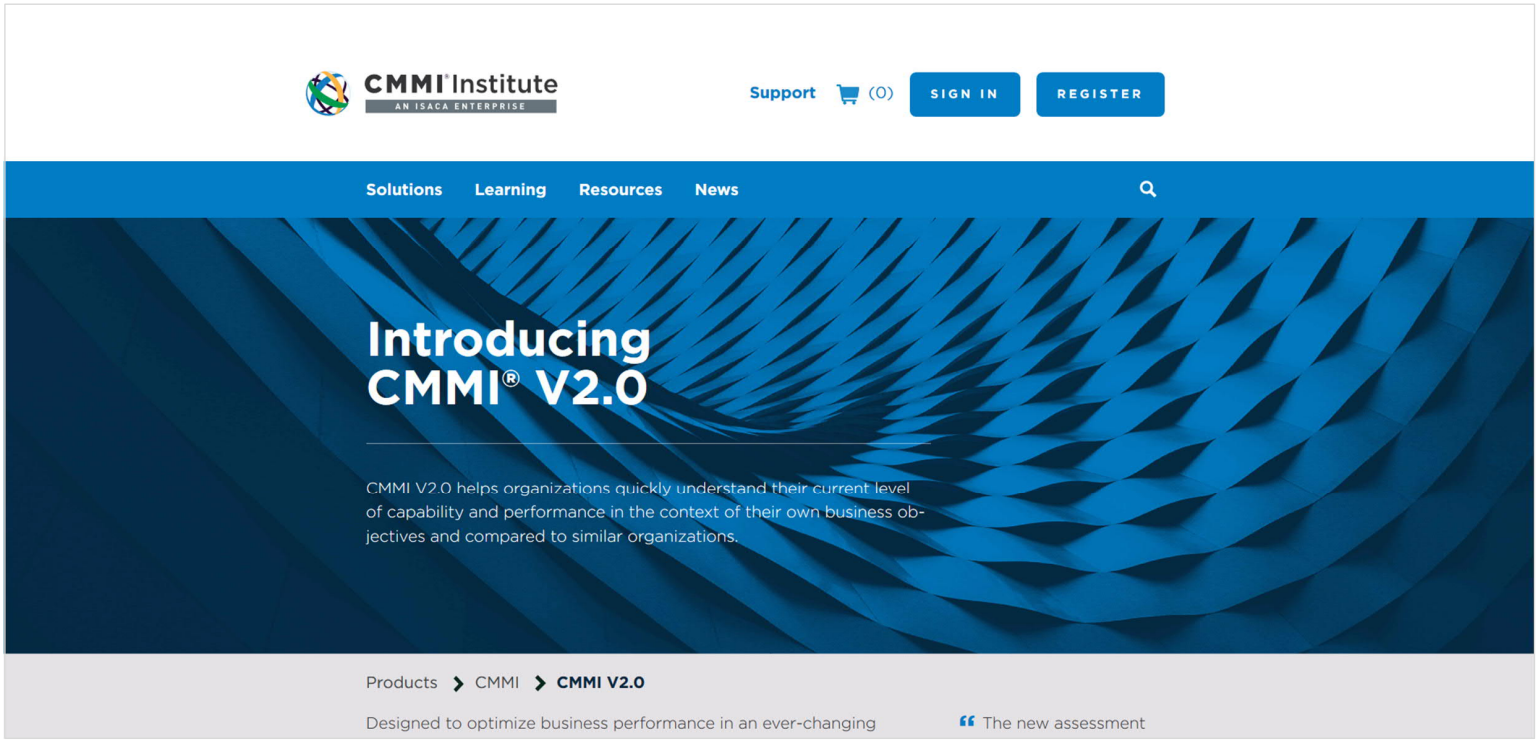
Figure 2.2 The Five Levels of Software Process Maturity

Tampereen yliopisto
Tampere University

CMMI levels



[cmmiinstitute.zendesk.com]



CMMI Institute cmmiinstitute.com/

The Software Capability Maturity Model (CMM) is a maturity framework for evaluating and improving the software development process. Carnegie Mellon University's (CMU) Software Engineering Institute (SEI) developed the model.

It is now managed by the CMMI Institute, ”a 100%-controlled subsidiary of Carnegie Innovations, Carnegie Mellon University’s technology commercialization enterprise.”

The goal of CMM is to develop a methodical framework for creating quality software that allows measurable and repeatable results.

Tampereen yliopisto
Tampere University

CMMI Levels of Capability and Performance

The maturity level or capability level of an organization provides a way to characterize its capability and performance. Experience has shown that organizations do their best when they focus their process improvement efforts on a prioritized and manageable number of practice areas at a time.

Capability levels apply to an organization's performance and process improvement achievements in individual practice areas.

Within practice areas, the practices are organized into practice groups labeled Level 0 to Level 5 which provide an evolutionary path to performance improvement.

Each level builds on the previous levels by adding new functionality or rigor resulting in increased capability.

Tampereen yliopisto
Tampere University

CMMI levels of Capability

Capability Level 0: Incomplete

- Incomplete approach to meeting the intent of the Practice Area.
- May or may not be meeting the intent of any practice.
- Inconsistent performance.

Capability Level 1: Initial

- Initial approach to meeting the intent of the Practice Area.
- Not a complete set of practices to meeting the full intent of the Practice Area.
- Addresses performance issues.

Capability Level 2: Managed

- Subsumes level 1 practices.
- Simple, but complete set of practices that address the full intent of the Practice Area.
- Does not require the use of the organizational assets.
- Identifies and monitors progress towards project performance objectives.

Capability Level 3: Defined

- Builds on level 2 practices.
- Uses organizational standards and tailoring to address project and work characteristics.
- Projects use and contribute to organization assets.
- Focuses on achieving both project and organizational performance objectives.

CMMI maturity levels [https://cmmiinstitute.com/learning/appraisals/levels]

Maturity Level 0: Incomplete

Ad hoc and unknown. Work may or may not get completed.

Maturity Level 1: Initial

Unpredictable and reactive. Work gets completed but is often delayed and over budget.

Maturity Level 2: Managed

Managed on the project level. Projects are planned, performed, measured, and controlled.

Maturity Level 3: Defined

Proactive, rather than reactive. Organization-wide standards provide guidance across projects, programs, and portfolios.

Maturity Level 4: Quantitatively Managed

Measured and controlled. Organization is data-driven with quantitative performance improvement objectives that are predictable and align to meet the needs of internal and external stakeholders.

Maturity Level 5: Optimizing

Stable and flexible. Organization is focused on continuous improvement and is built to pivot and respond to opportunity and change. The organization's stability provides a platform for agility and innovation.

(Quality) certificates and certification

Tampereen yliopisto
Tampere University

Organisation's quality certification

You need to to apply for some quality certificate only because of business reasons. Usually customers require some quality certificate from developer company (e.g. ISO 9001, AQAP 2110,...).

- first you need to modify your process
- then you write a quality handbook
- make sure every worker understands the guidelines and follows those
- take a quality audit (that is expensive)
- if not pass, correct and re-audit
- keep process and documentation updated
- annual renewals (which are expensive).

Tampereen yliopisto
Tampere University

Costs of ISO 9001 Certification

The cost of ISO 9001 certification can make organizations reluctant to become certified. Although becoming ISO 9001 certified can be expensive, there are many factors that influence the price of the overall project, of which many are in your control. While there are costs associated with implementing ISO such as the cost of a registrar and internal resource cost, you should view certification as an investment in the organization's growth.

Size and complexity of your organization will greatly influence the cost of implementation. One way to help control the size can be to limit the scope. While this should always be done for the benefit of the business's intended outcomes, companies may be mindful that the scope directly impacts the cost of Certification. For example, additional branches and locations require additional audit days, so the benefits of including branch offices should be carefully considered.

[the9000store.com/articles/iso-9000-cost/]

Tampereen yliopisto
Tampere University

26-100 Employees, no quality system in place yet

Estimated costs of doing the implementation in-house yourself using our All-in-One Documentation & Training Package verses hiring a consultant.

	Cost in **Dollars** for All-in-One Package or Consultant	Cost in terms of your company's employee **hours** spent
All-in-One	$997	288*
Consultant	$11,250*	144*

the9000store.com/articles/iso-9000-cost/

26-100 Employees, basic quality system in place

Estimated costs of doing the implementation in-house yourself using our All-in-One Documentation & Training Package verses hiring a consultant.

	Cost in **Dollars** for All-in-One Package or Consultant	Cost in terms of your company's employee **hours** spent
All-in-One	$997	216*
Consultant	$8,125*	108*

**These are estimates.*

Remember to add the cost of the Registration Audit to your total cost estimate.

26-100 Employees, good quality system in place

Estimated costs of doing the implementation in-house yourself using our All-in-One Documentation & Training Package verses hiring a consultant.

	Cost in **Dollars** for All-in-One Package or Consultant	Cost in terms of your company's employee **hours** spent
All-in-One	$997	115*
Consultant	$5,000*	58*

**These are estimates.*

Remember to add the cost of the Registration Audit to your total cost estimate.

Tampereen yliopisto
Tampere University

251-500 Employees, no quality system in place yet

Estimated costs of doing the implementation in-house yourself using our All-in-One Documentation & Training Package verses hiring a consultant.

	Cost in **Dollars** for All-in-One Package or Consultant	Cost in terms of your company's employee **hours** spent
All-in-One	$997	960*
Consultant	$41,250*	480*

the9000store.com/articles/iso-9000-cost/

251-500 Employees, basic quality system in place

Estimated costs of doing the implementation in-house yourself using our All-in-One Documentation & Training Package verses hiring a consultant.

	Cost in **Dollars** for All-in-One Package or Consultant	Cost in terms of your company's employee **hours** spent
All-in-One	$997	720*
Consultant	$28,125*	360*

**These are estimates.*

Remember to add the cost of the Registration Audit to your total cost estimate.

251-500 Employees, good quality system in place

Estimated costs of doing the implementation in-house yourself using our All-in-One Documentation & Training Package verses hiring a consultant.

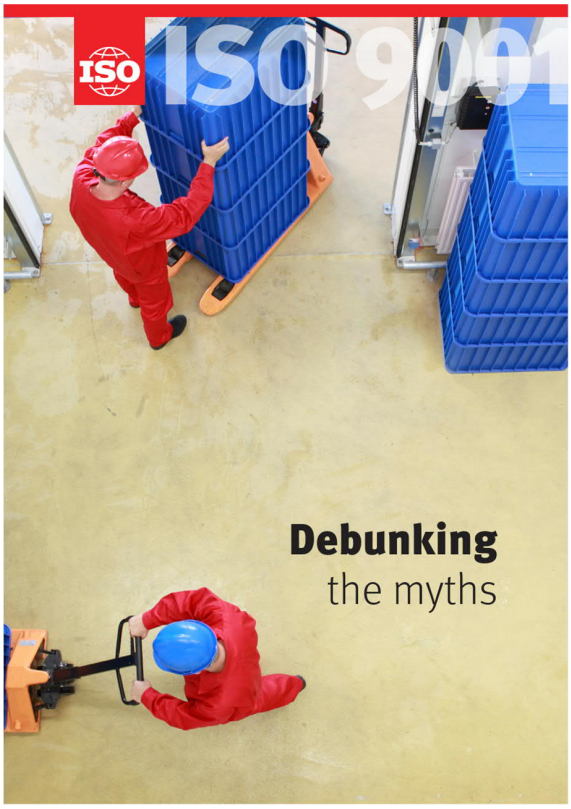
	Cost in **Dollars** for All-in-One Package or Consultant	Cost in terms of your company's employee **hours** spent
All-in-One	$997	384*
Consultant	$15,000*	192*

**These are estimates.*

Remember to add the cost of the Registration Audit to your total cost estimate.

ISO QMP

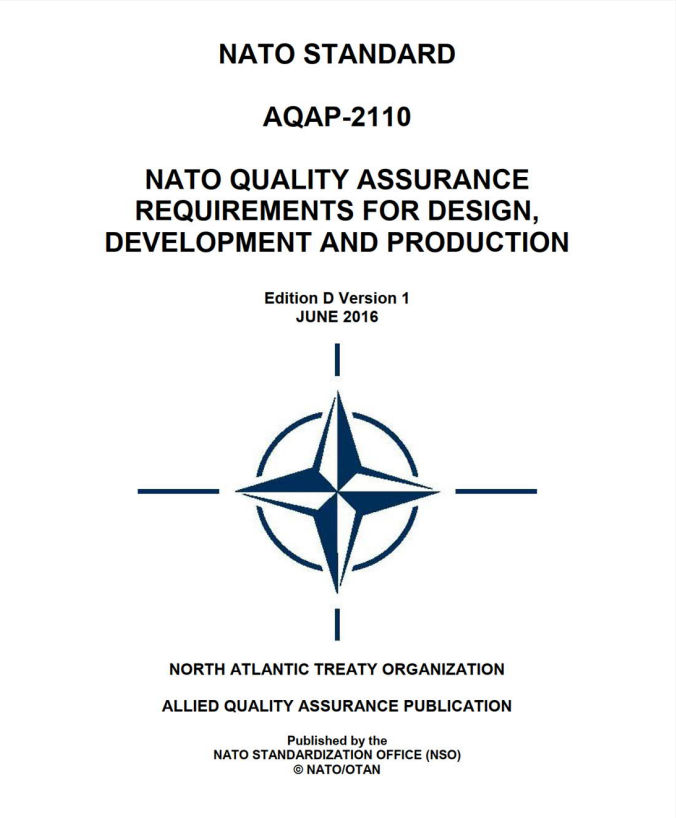
(quality management principles)



Tick IT [https://www.tickitplus.org/]

TickITplus offers a flexible, multi-level approach to IT quality and certification assessment and can be applied at whatever level is deemed appropriate to the quality and process maturity of the organization and the needs of its customers. While ISO 9001 is the mandatory standard, TickITplus can also provide certification for other IT Standards under one certification arrangement.

The Allied Quality Assurance Publications (AQAP) are standards for quality assurance systems that have been developed by NATO.
The aim of the AQAP agreement is to define standards for Quality Assurance of defence products. These standards are an integral part of contracts awarded in the military field involving NATO member countries. AQAP documents are therefore important to contractors and companies wanting to bid for such contracts. The AQAP system is described in STANAG 4107 issued by the NATO Standardization Agency.



NATO STANDARD

AQAP-2110

NATO QUALITY ASSURANCE REQUIREMENTS FOR DESIGN, DEVELOPMENT AND PRODUCTION

Edition D Version 1
JUNE 2016

NORTH ATLANTIC TREATY ORGANIZATION

ALLIED QUALITY ASSURANCE PUBLICATION

Published by the
NATO STANDARDIZATION OFFICE (NSO)
© NATO/OTAN

AQAP series structure

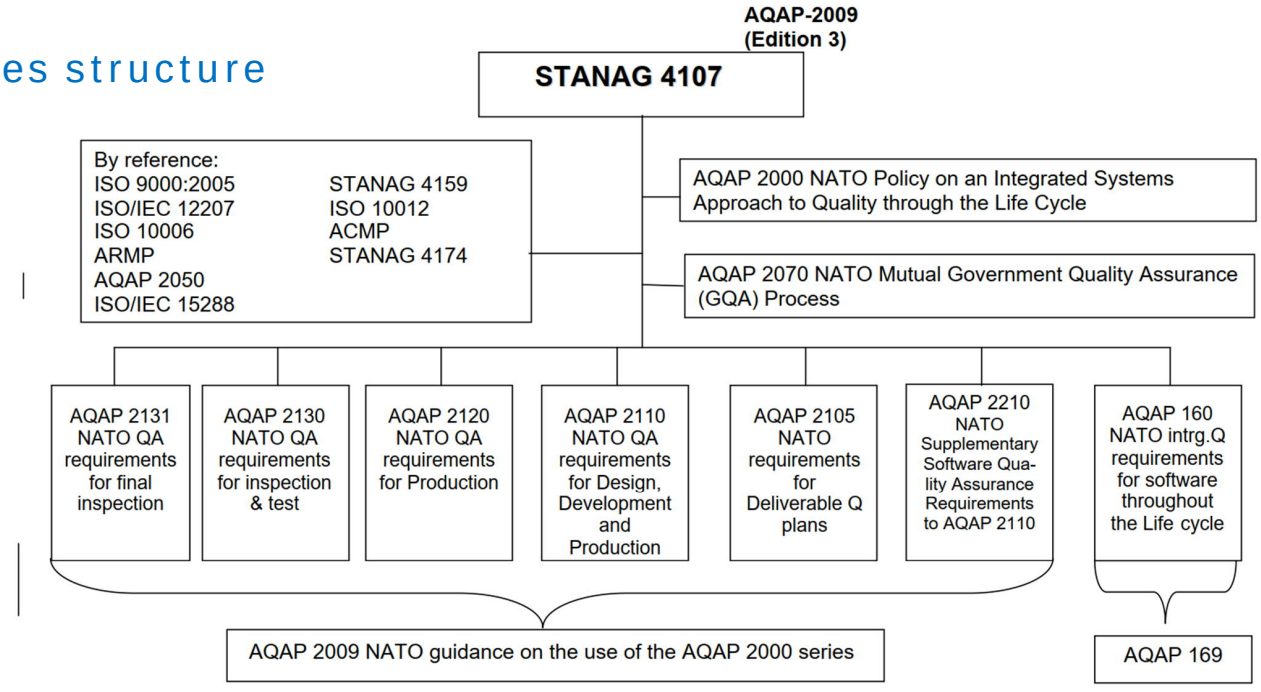


Figure 1

Tampereen yliopisto
Tampere University

auditing (ISO 24765-2017)

3.298 , audit

1. systematic, independent, documented process for obtaining records, statements of fact, or other relevant information and assessing them objectively, to determine the extent to which specified requirements are fulfilled

2. independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria

3. independent examination of a software product, software process, or set of software processes to assess compliance with specifications, standards, contractual agreements, or other criteria

4. independent assessment of products and processes, conducted by an authorized person to assess compliance with requirements

5. systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which audit criteria are fulfilled.

Note 1 to entry: An audit results in a clear indication of whether the audit criteria have been met.

3.299 , audit team

1. one or more auditors conducting an audit, supported if needed by technical experts.

Note 1 to entry: One auditor of the audit team is appointed as the audit team leader. The audit team can include auditors-intraining.

Tampereen yliopisto
Tampere University

Tampereen yliopisto
Tampere University

Highlights - What to remember

- embedded software systems must keep both sw and hw development in control
- ”good code comes only by following the rules”
- use state diagrams for design and development help
- a lot of hype at IoT and AI at the moment; few experts available
- many Finnish and European Union projects started and ongoing about IoT and AI
- cyber security needs consideration in all software development, and especially IoT
- network with other companies/organisations, you do not know everything yourself
- by benchmarking you may find out your organisation’s level and areas to improve
-

- university courses about IoT and AI; at TUNI Spring 2020

Tampereen yliopisto
Tampere University

Tampereen yliopisto
Tampere University

Now the additional L12 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Tampereen yliopisto
Tampere University

Now the additional L12 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Tampereen yliopisto
Tampere University

Now the additional L12 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Tampereen yliopisto
Tampere University

Now the additional L12 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Tampereen yliopisto
Tampere University

Embedded systems

Characteristics of an Embedded System

• Single-functioned – An embedded system usually performs a specialized operation and does the same repeatedly. For example: A pager always functions as a pager.

• Tightly constrained – All computing systems have constraints on design metrics, but those on an embedded system can be especially tight. Design metrics is a measure of an implementation's features such as its cost, size, power, and performance. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.

• Reactive and Real time – Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay. Consider an example of a car cruise controller; it continually monitors and reacts to speed and brake sensors. It must compute acceleration or de-accelerations repeatedly within a limited time; a delayed computation can result in failure to control of the car.

• Microprocessors based – It must be microprocessor or microcontroller based.

• Memory – It must have a memory, as its software usually embeds in ROM. It does not need any secondary memories in the computer.

• Connected – It must have connected peripherals to connect input and output devices.

• HW-SW systems – Software is used for more features and flexibility. Hardware is used for performance and security.

[https://www.tutorialspoint.com/embedded_systems/]

Terminologies used in embedded system

Reliability: This measure of the survival probability of the system when the function is critical during the run time.

Fault-Tolerance: Fault-Tolerance is the capability of a computer system to survive in the presence of faults.

Real-Time: Embedded system must meet various timing and other constraints. They are imposed on it by the real-time natural behavior of the external world.

For example, an airforce department which keeps track of incoming missile attacks must precisely calculate and plan their counter-attack due to hard real-time deadline. Otherwise, it'll get destroyed.

Flexibility: It's building systems with built-in debugging opportunities which allows remote maintenance.

For example, you are building a spacecraft which will land on another planter to collect various types of data and send collected detail back to us. If this spacecraft went insane and lost the control, we should be able to make some important diagnostic. So, flexibility is vital while designing an embedded system.

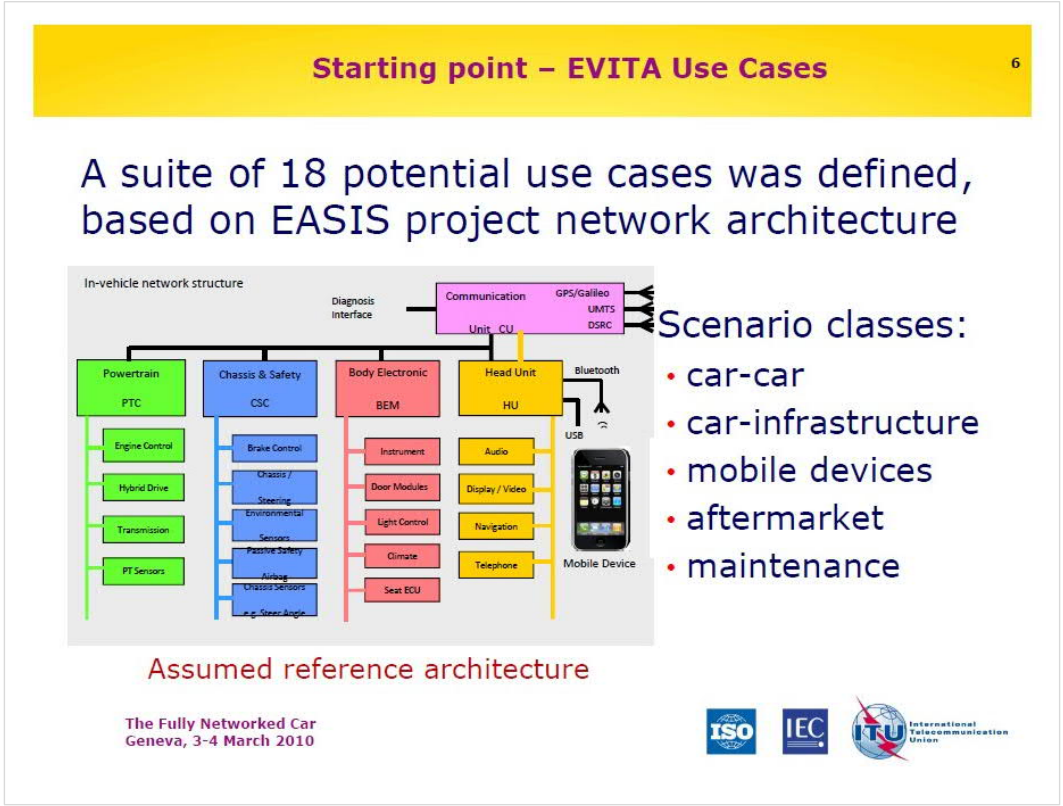
Portability: Portability is a measure of the ease of using the same embedded software in various environments. It requires generalized abstractions between the application program logic itself and the low-level system interfaces.

[https://www.guru99.com/embedded-systems-tutorial.html]

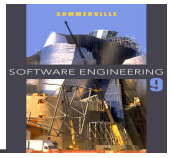
Embedded systems in short [www.guru99.com/embedded-systems-tutorial.html]

- Embedded systems means a combination of computer software and hardware which is either fixed in capability or programmable.
- Embedded system requires real time performance.
- Reliability measure of the survival probability of the system when the function is critical during the run time.
- Fault-Tolerance is the capability of a computer system to survive in the presence of faults.
- Embedded system must meet various timing and other constraints.
- Flexibility is building systems with built-in debugging opportunities which allows remote maintenance.
- Portability is a measure of the ease of using the same embedded software in various environments.
- A microcontroller is a single-chip VLSI unit which is also called microcomputer.
- Application of Embedded System includes: 1) Robotic science, 2) Medical, 3) Automotive, 3) Networking, 4)Home Devices, 5) Automobiles, and 6) Industrial Control.
- Major advantages of Embedded System is that it is able to cover a wide variety of environments.
- The major drawback of Embedded System is that it needs a long time to market.

Security and safety risks in cars



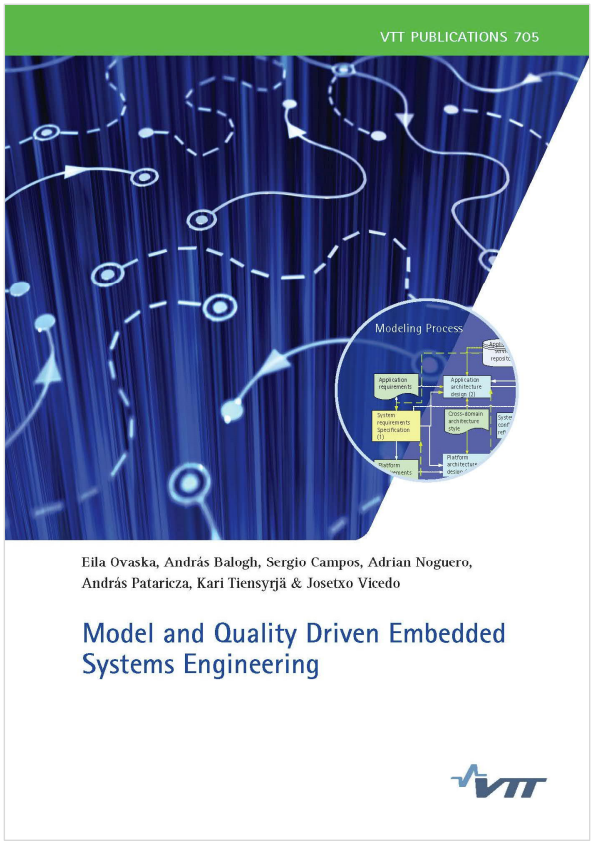
Architectural considerations



- Because of the need to respond to timing demands made by different stimuli/responses, the **system architecture must allow for fast switching between stimulus handlers**.
- **Timing demands** of different stimuli are different so a simple sequential loop is not usually adequate.
- Real-time systems are therefore usually designed as cooperating processes with a real-time executive controlling these processes.

On the other hand, too fast pushing on the buttons may cause crashes, like in older TV remote controllers. There were also undocumented commands (also in older GSM phones… well, almost all devices have such, for testing and maintenance).

Tampereen yliopisto
Tampere University



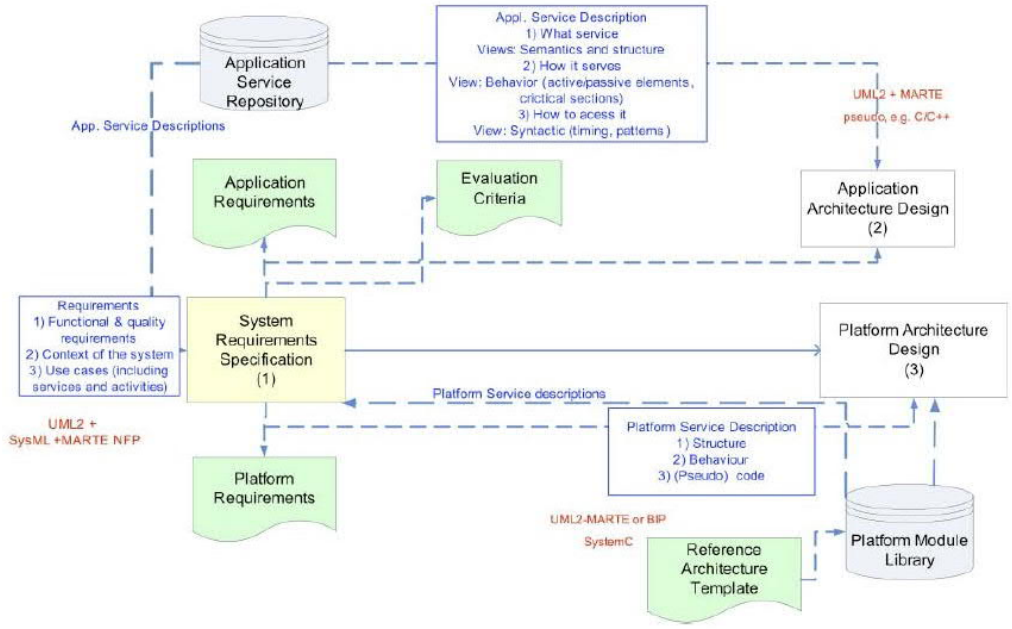


Figure 4. Overview of the System Requirements Specification phase.

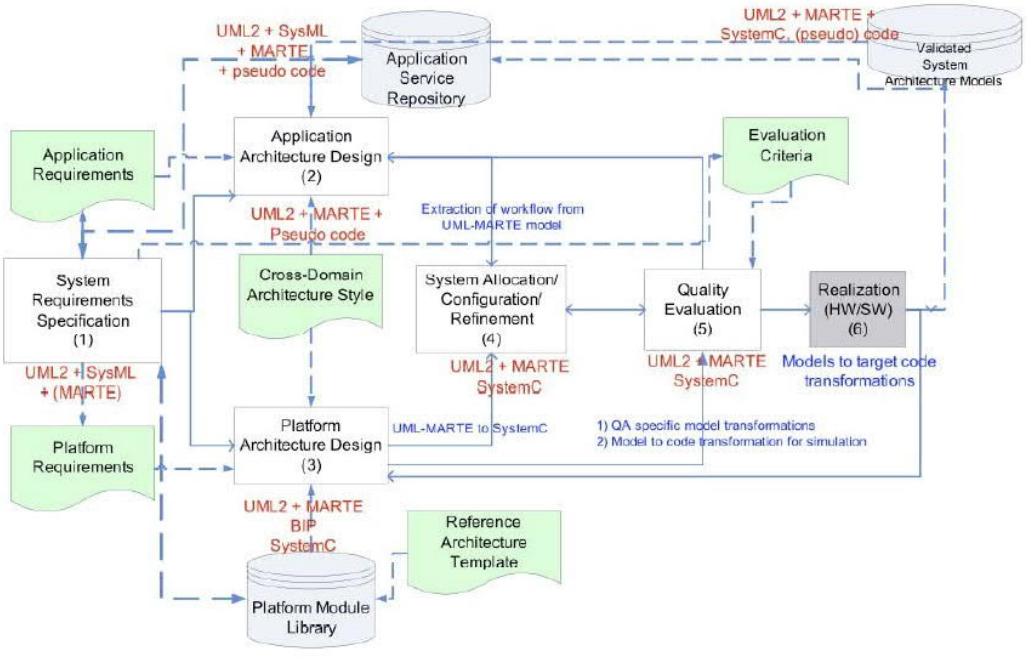
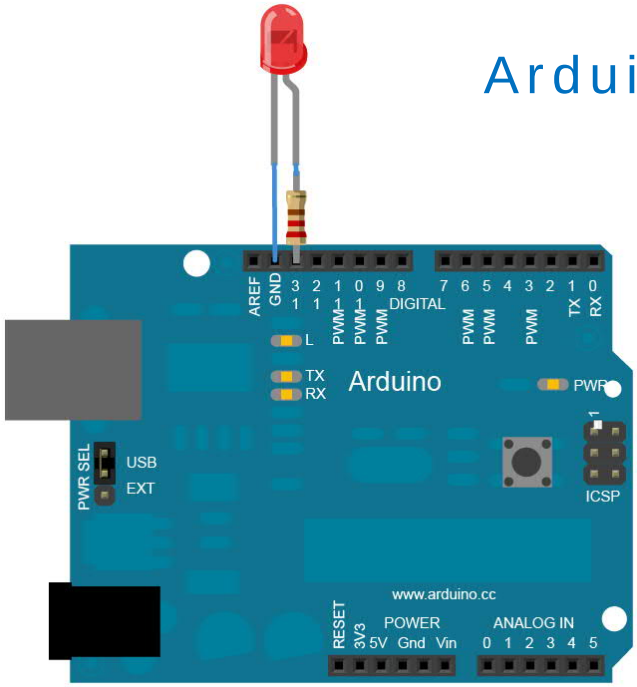
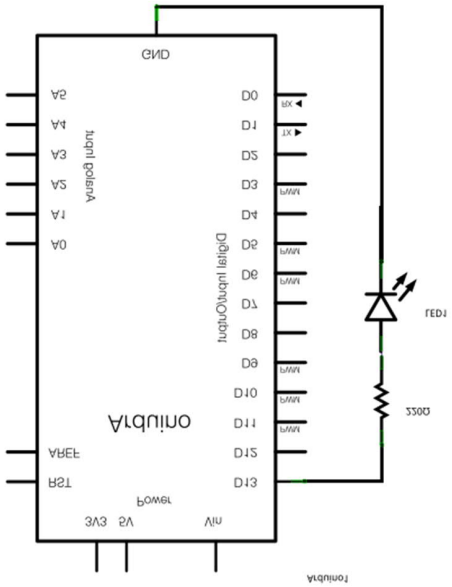


Figure 16. Architecture design phases and the selected modelling languages.

Arduino





Best 52 single-board computers, 2019: [https://www.slant.co/topics/1629/~best-single-board-computers]

Tampereen yliopisto
Tampere University

eCos = embedded configurable operating system

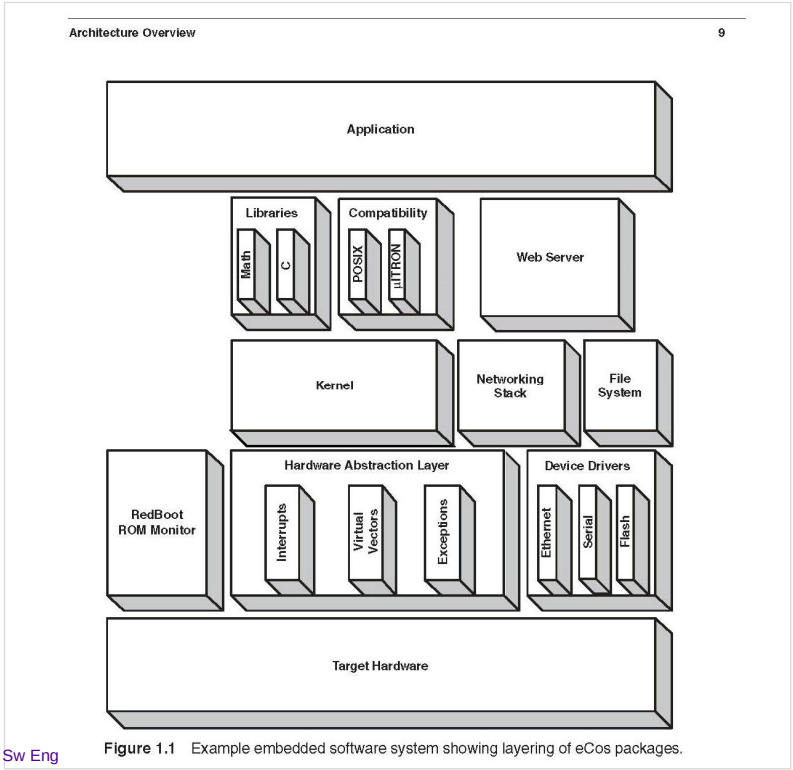


Figure 1.1 Example embedded software system showing layering of eCos packages.

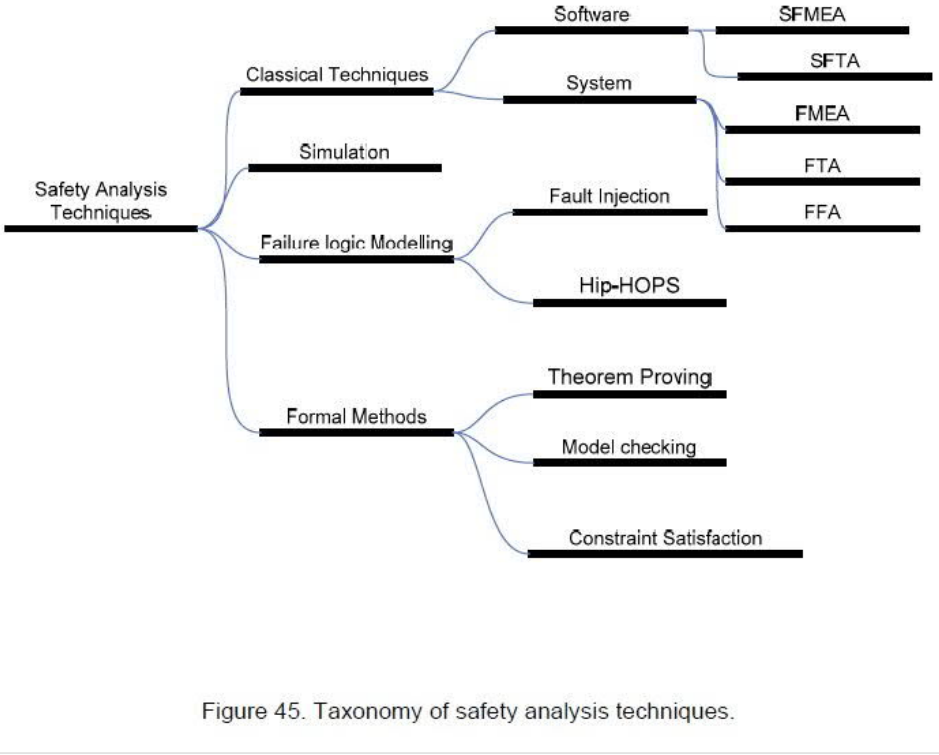
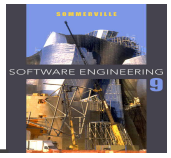


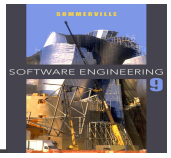
Figure 45. Taxonomy of safety analysis techniques.

Responsiveness



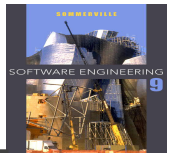
- Responsiveness in real-time is the critical difference between embedded systems and other software systems, such as information systems, web-based systems or personal software systems.
- For non-real-time systems, correctness can be defined by specifying how system inputs map to corresponding outputs that should be produced by the system.
- In a real-time system, the correctness depends both on the response to an input and the time taken to generate that response. If the system takes too long to respond, then the required response may be ineffective.

Embedded software



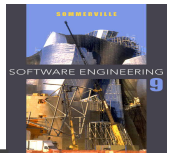
- Computers are used to **control** a wide range of systems from simple domestic machines, through games controllers, to entire manufacturing plants.
- Their software must **react to events** generated by the hardware and, often, issue control signals in response to these events.
- The **software in these systems is embedded in system hardware**, often in read-only memory, and usually **responds**, in real time, to events from the system's environment.

Definitions



- A real-time system is a software system where the correct functioning of the system depends on the results produced by the system and the time at which these results are produced.
- A soft real-time system is a system whose operation is degraded if results are not produced according to the specified timing requirements.
- A hard real-time system is a system whose operation is incorrect if results are not produced according to the timing specification.

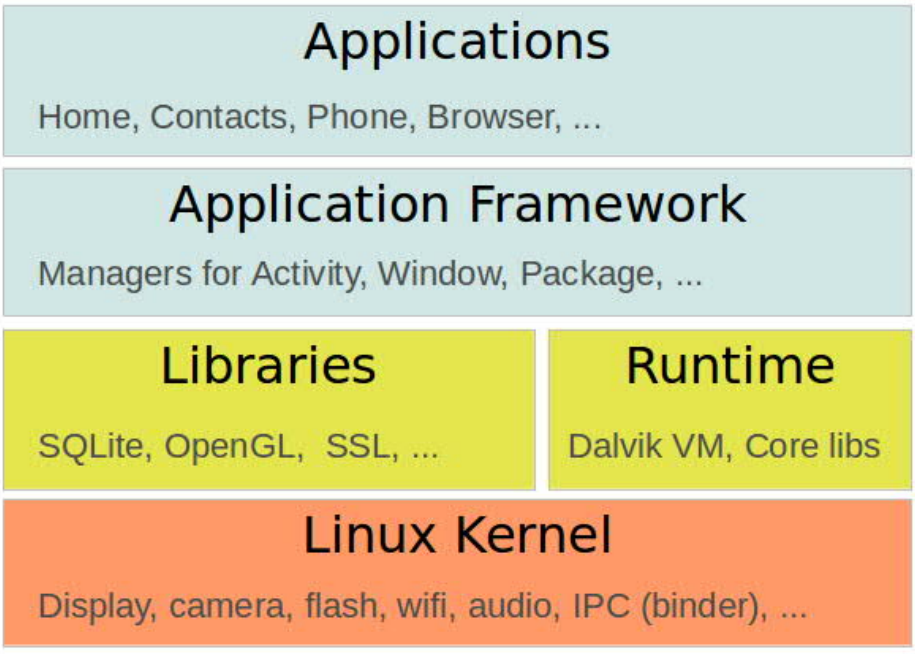
Characteristics of embedded systems



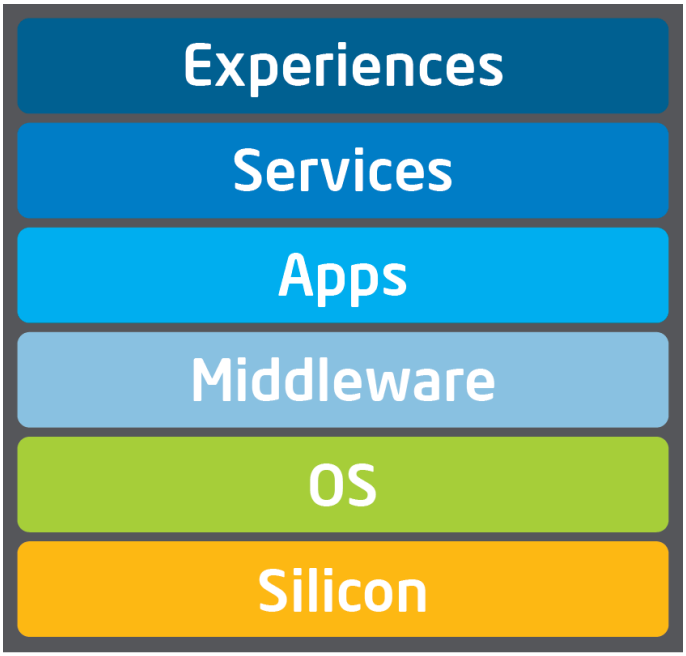
- ✧ Embedded systems generally run continuously and do not terminate.
- ✧ Interactions with the system's environment are unpredictable.
- ✧ There may be physical limitations that affect the design of a system.
- ✧ Direct hardware interaction may be necessary.
- ✧ Issues of safety and reliability may dominate the system design.

Tampereen yliopisto
Tampere University

Android software stack



[https://dzone.com/articles/android-software-stack-and]



Adapted by Intel IT Center from Genevieve Bell's IDF 2013 Keynote Address

[https://itpeernetwork.intel.com/software-innovation-for-a-better-mobile-experience/#gs.h87q42]

eCos = embedded configurable operating system

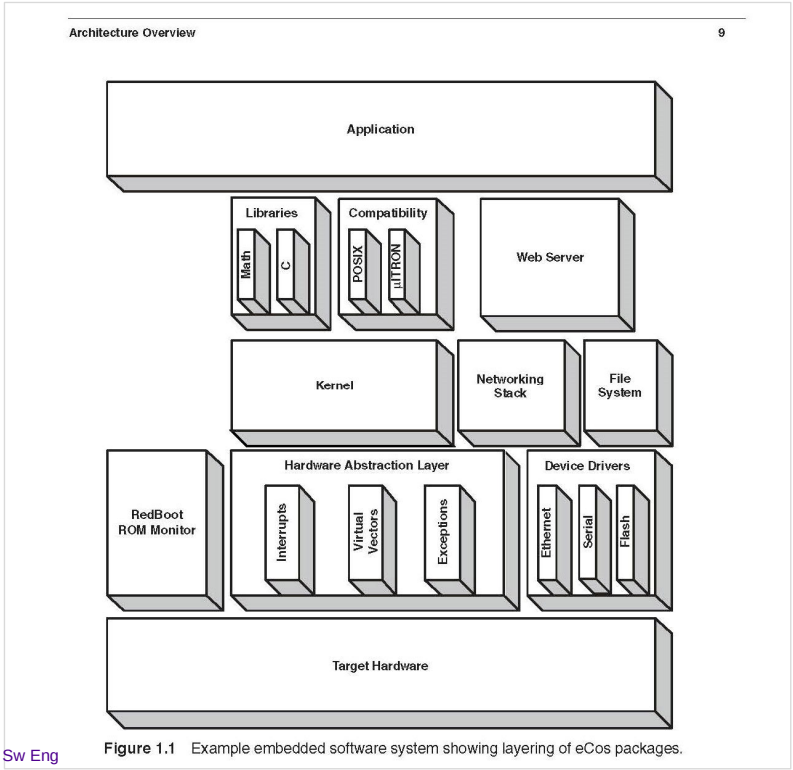
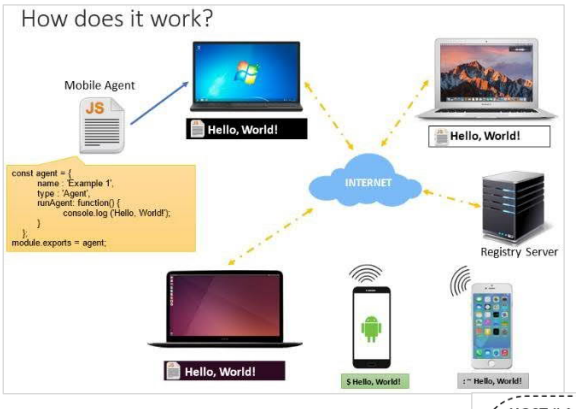
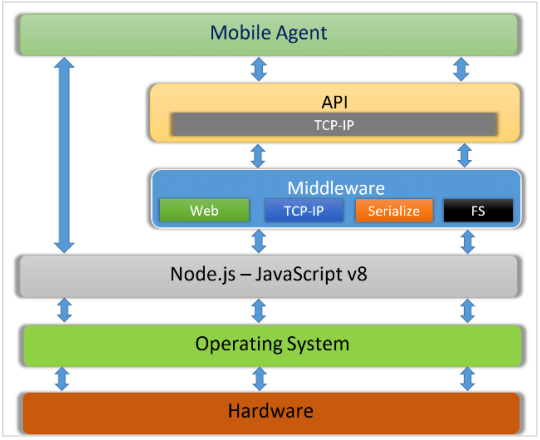
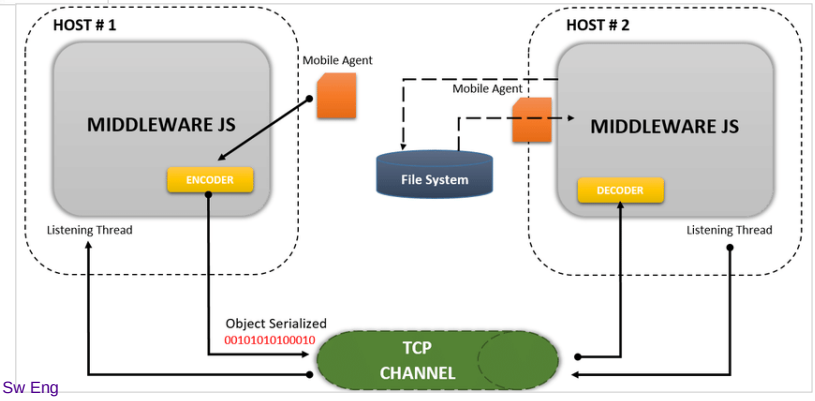


Figure 1.1 Example embedded software system showing layering of eCos packages.

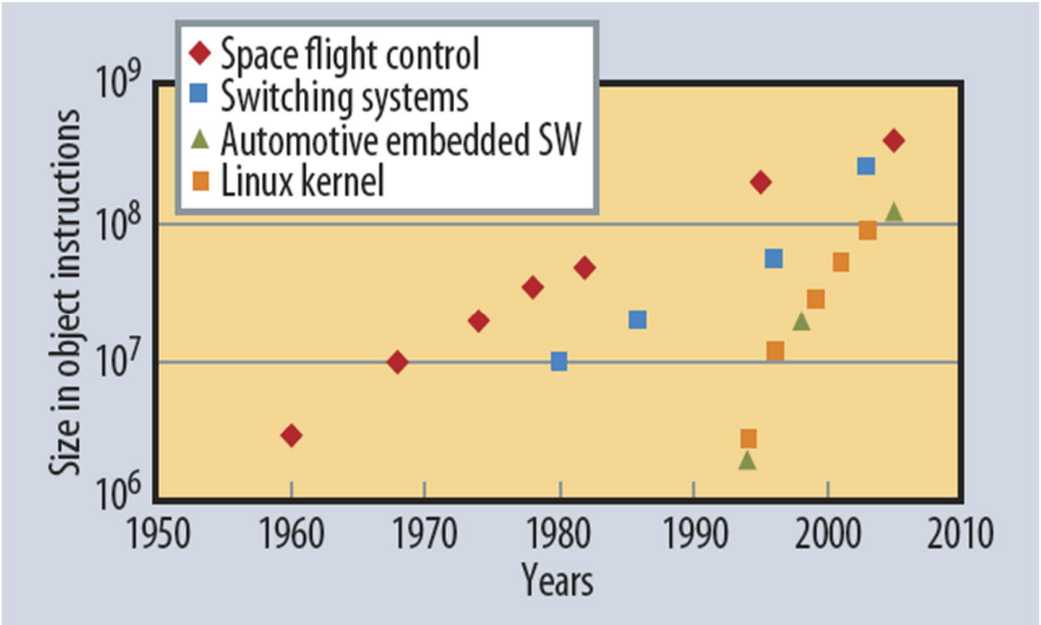




[Carlos Villafuerte: JavaScript Middleware for Mobile Agents Support on Desktop and Mobile Platforms, 2018]

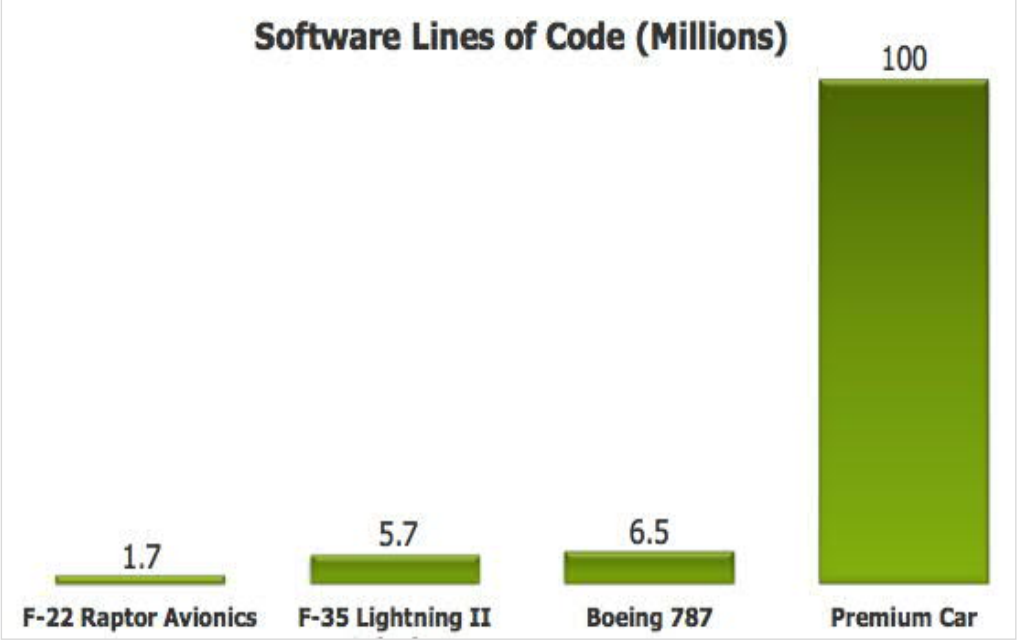


Software size, increasing all the time



Embedded Software: Facts, Figures, and Future IEEE Computer, April 2009 (vol. 42 no. 4)

Some LOC counts

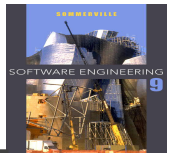


Tampereen yliopisto
Tampere University

Tampereen yliopisto
Tampere University

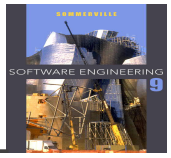
RT, Real-time systems

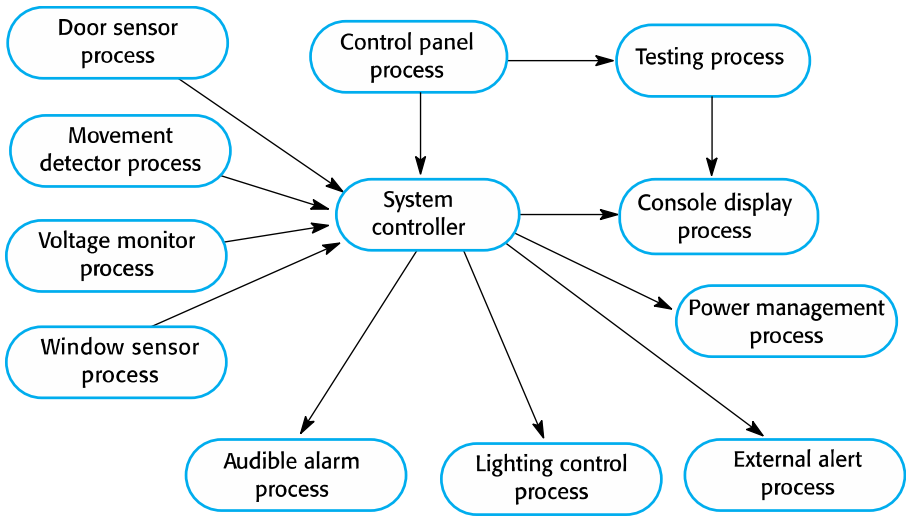
Real-time programming



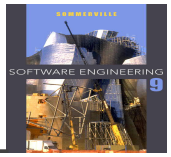
- Programming languages for real-time systems development have to include facilities to access system hardware, and it should be possible to predict the timing of particular operations in these languages.
- Systems-level languages, such as C, which allow efficient code to be generated are widely used in preference to languages such as Java.
- There is a performance overhead in object-oriented systems because extra code is required to mediate access to attributes and handle calls to operations. The loss of performance may make it impossible to meet real-time deadlines.

Process structure for a burglar alarm system



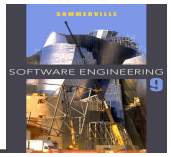


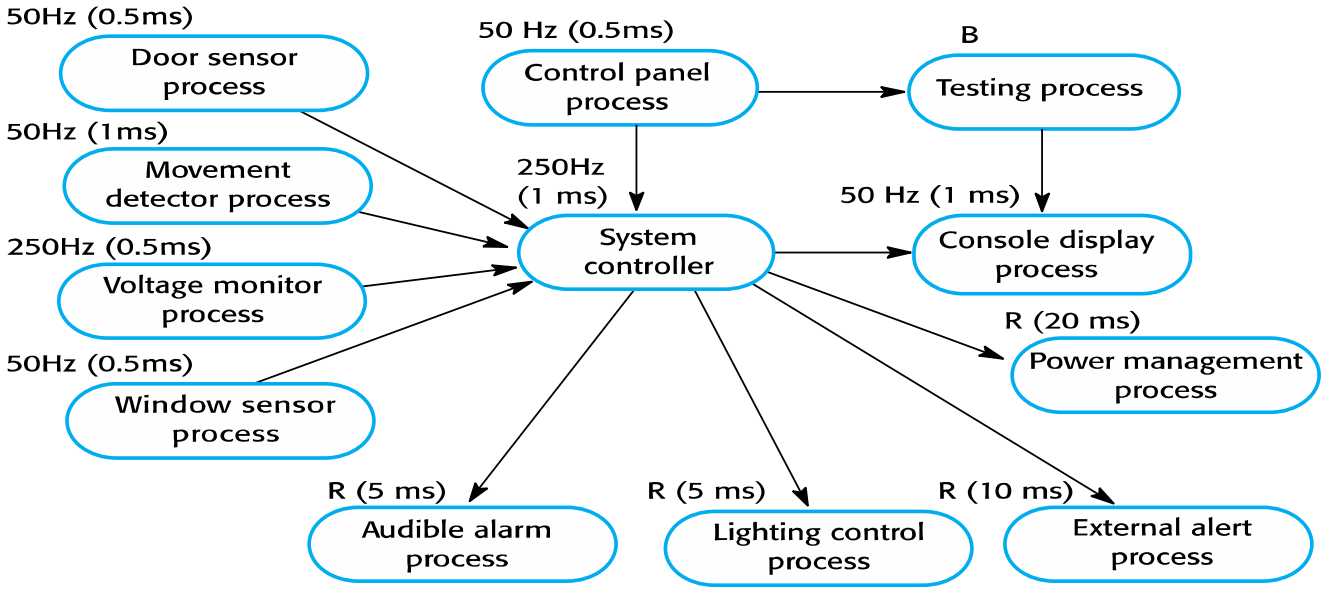
Timing analysis



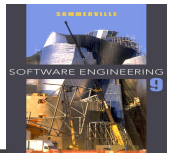
- The correctness of a real-time system depends not just on the correctness of its **outputs** but also on the **time** at which these outputs were produced.
- In a timing analysis, you calculate how often each process in the system must be executed to ensure that all inputs are processed and all system responses produced in a timely way.
- The results of the timing analysis are used to decide how frequently each process should execute and how these processes should be scheduled by the real-time operating system.

Alarm process timing



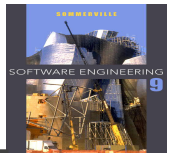


Factors in timing analysis



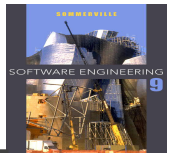
- ***Deadlines***
  - The times by which stimuli must be processed and some response produced by the system.
- ***Frequency***
  - The number of times per second that a process must execute so that you are confident that it can always meet its deadlines.
- ***Execution time***
  - The time required to process a stimulus and produce a response.

Stimuli to be processed



- Power failure is detected by observing a voltage drop of more than 20%.
  - The required response is to switch the circuit to backup power by signalling an electronic power-switching device that switches the mains power to battery backup.
- Intruder alarm is a stimulus generated by one of the system sensors.
  - The response to this stimulus is to compute the room number of the active sensor, set up a call to the police, initiate the voice synthesizer to manage the call, and switch on the audible intruder alarm and building lights in the area.

Frequency and execution time

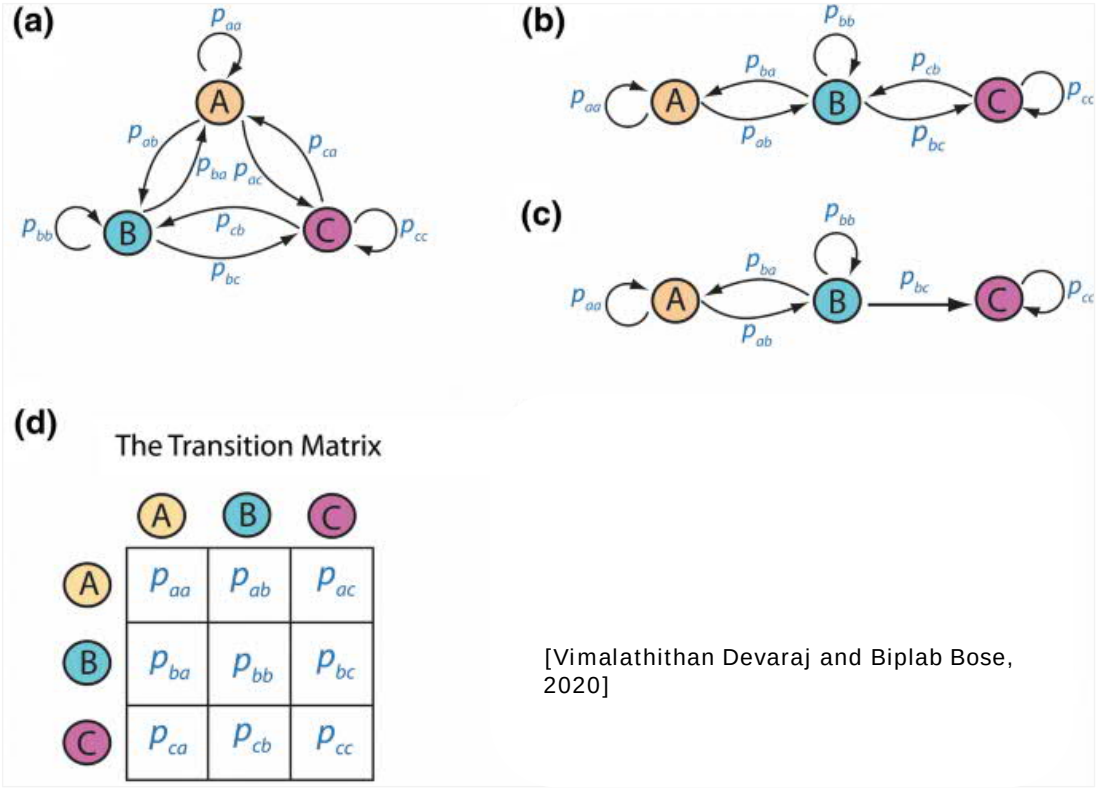


- The deadline for detecting a change of state is 0.25 seconds, which means that each sensor has to be checked 4 times per second. If you examine 1 sensor during each process execution, then if there are N sensors of a particular type, you must schedule the process 4N times per second to ensure that all sensors are checked within the deadline.
- If you examine 4 sensors, say, during each process execution, then the execution time is increased to about 4 ms, but you need only run the process N times/second to meet the timing requirement.

Safety

State transition matrix

(d)



[Vimalathithan Devaraj and Biplab Bose, 2020]

ISO 26262 series (2019)

- SFS-ISO 26262-1:2019, Road vehicles -- Functional safety -- Part 1: Vocabulary
- SFS-ISO 26262-2:2019, Road vehicles -- Functional safety -- Part 2: Management of functional safety
- SFS-ISO 26262-3:2019, Road vehicles -- Functional safety -- Part 3: Concept phase
- SFS-ISO 26262-4:2019, Road vehicles -- Functional safety -- Part 4: Product development at the system level
- SFS-ISO 26262-5:2019, Road vehicles -- Functional safety -- Part 5: Product development at the hardware level
- SFS-ISO 26262-6:2019, Road vehicles -- Functional safety -- Part 6: Product development at the software level
- SFS-ISO 26262-7:2019, Road vehicles -- Functional safety -- Part 7: Production, operation, service and decommissioning
- SFS-ISO 26262-8:2019, Road vehicles -- Functional safety -- Part 8: Supporting processes
- SFS-ISO 26262-9:2019, Road vehicles -- Functional safety -- Part 9: Automotive safety integrity level (ASIL)-oriented and safety-oriented analyses
- SFS-ISO 26262-10:2019, Road vehicles -- Functional safety -- Part 10: Guidelines on ISO 26262
- SFS-ISO 26262-11:2019, Road vehicles -- Functional safety -- Part 11: Guidelines on application of ISO 26262 to semiconductors
- SFS-ISO 26262-12:2019, Road vehicles -- Functional safety -- Part 12: Adaptation of ISO 26262 for motorcycles.

Standards (ISO 14229)

- ISO 14229-1:2013 Road vehicles -- Unified diagnostic services (UDS) -- Part 1: Specification and requirements
- ISO 14229-2:2013 Road vehicles -- Unified diagnostic services (UDS) -- Part 2: Session layer services
- ISO 14229-3:2012 Road vehicles -- Unified diagnostic services (UDS) -- Part 3: Unified diagnostic services on CAN implementation (UDSonCAN)
- ISO 14229-4:2012 Road vehicles -- Unified diagnostic services (UDS) -- Part 4: Unified diagnostic services on FlexRay implementation (UDSonFR)
- ISO 14229-5:2013 Road vehicles -- Unified diagnostic services (UDS) -- Part 5: Unified diagnostic services on Internet Protocol implementation (UDSonIP)
- ISO 14229-6:2013 Road vehicles -- Unified diagnostic services (UDS) -- Part 6: Unified diagnostic services on K-Line implementation (UDSonK-Line)

Standards (ISO 11992)

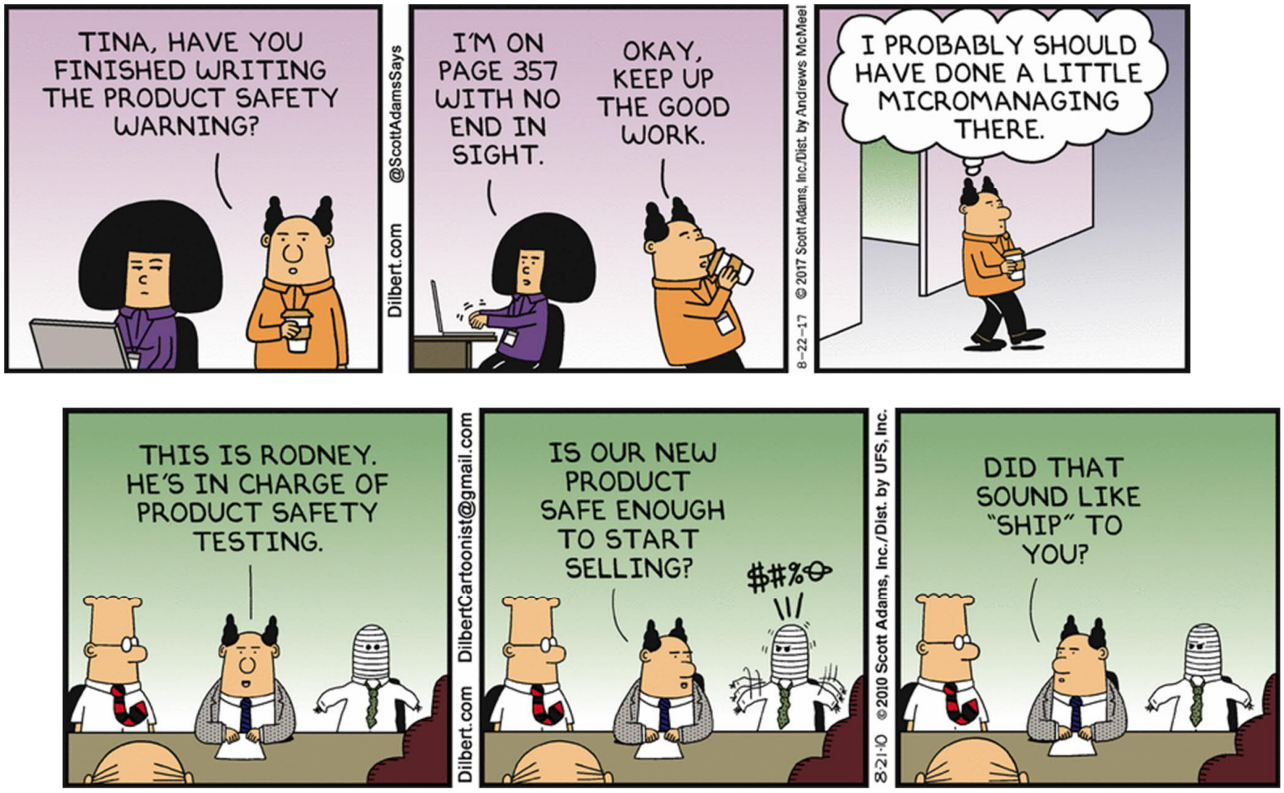
- ISO 11992-1:2003 Road vehicles -- Interchange of digital information on electrical connections between towing and towed vehicles -- Part 1: Physical and data-link layers
- ISO 11992-2:2014 Road vehicles -- Interchange of digital information on electrical connections between towing and towed vehicles -- Part 2: Application layer for brakes and running gear
- ISO 11992-3:2003 Road vehicles -- Interchange of digital information on electrical connections between towing and towed vehicles -- Part 3: Application layer for equipment other than brakes and running gear
- ISO 11992-4:2014 Road vehicles -- Interchange of digital information on electrical connections between towing and towed vehicles -- Part 4: Diagnostic communication

Standards (ISO 17356)

- ISO 17356-1:2005 Road vehicles -- Open interface for embedded automotive applications -- Part 1: General structure and terms, definitions and abbreviated terms
- ISO 17356-2:2005 Road vehicles -- Open interface for embedded automotive applications -- Part 2: OSEK/VDX specifications for binding OS, COM and NM
- ISO 17356-3:2005 Road vehicles -- Open interface for embedded automotive applications -- Part 3: OSEK/VDX Operating System (OS)
- ISO 17356-4:2005 Road vehicles -- Open interface for embedded automotive applications -- Part 4: OSEK/VDX Communication (COM)
- ISO 17356-5:2006 Road vehicles -- Open interface for embedded automotive applications -- Part 5: OSEK/VDX Network Management (NM)
- ISO 17356-6:2006 Road vehicles -- Open interface for embedded automotive applications -- Part 6: OSEK/VDX Implementation Language (OIL)

Standards 4

- ISO/IEC TR 18037:2008 Programming languages -- C -- Extensions to support embedded processors
- ISO 13849-1:2015 Safety of machinery -- Safety-related parts of control systems -- Part 1: General principles for design
- ISO 13849-2:2012 Safety of machinery -- Safety-related parts of control systems -- Part 2: Validation
- ISO 22201-2:2013 Lifts (elevators), escalators and moving walks -- Programmable electronic systems in safety related applications -- Part 2: Escalators and moving walks (PESSRAE)
- ISO/IEC 14762:2009 Information technology -- Functional safety requirements for Home and Building Electronic Systems (HBES)



Four views on software systems safety

DEPARTMENT OF DEFENSE

JOINT SOFTWARE SYSTEMS SAFETY ENGINEERING HANDBOOK

DEVELOPED BY THE JOINT SOFTWARE SYSTEMS SAFETY ENGINEERING WORKGROUP

Original published December 1999
Version 1.0 Published August 27, 2010

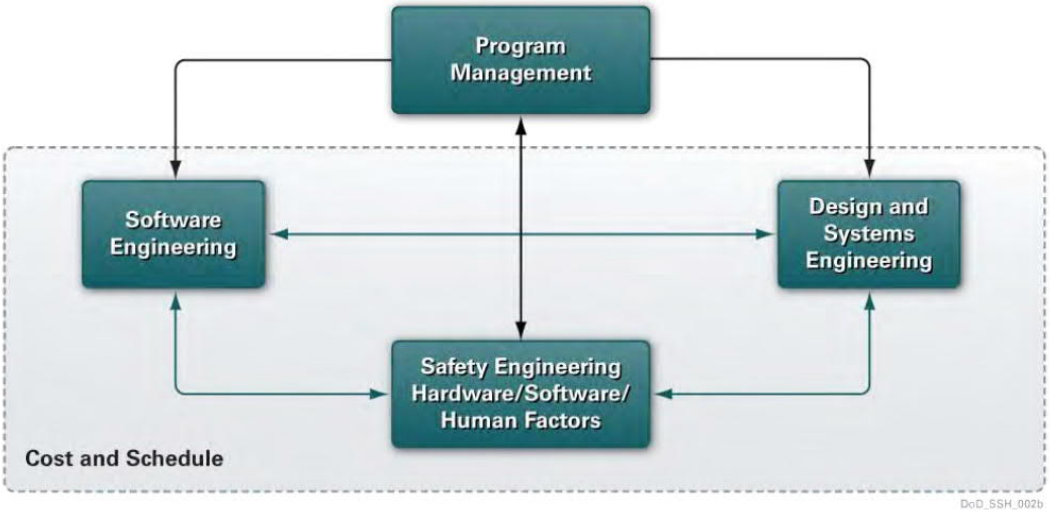


Figure 2-1: Management Commitment to the Integrated Safety Process

NATIONAL TRANSPORTATION SAFETY BOARD

Search this site... Search Site

Advanced Search

HOME NEWS & EVENTS SAFETY ADVOCACY INVESTIGATIONS DISASTER ASSISTANCE LEGAL ABOUT SEARCH NEW

PUBLICATIONS

Home > NEWS & EVENTS > News Releases > Driver Errors, Advanced Driver Assistance System Design, Led to Highway Crash

SHARE

NTSB News Release

National Transportation Safety Board Office of Public Affairs

Driver Errors, Advanced Driver Assistance System Design, Led to Highway Crash

9/4/2019

WASHINGTON (Sept. 4, 2019) — A driver's inattention, overreliance on his car's advanced driver assistance system, and use of the system inconsistent with manufacturer guidance, coupled with the system permitting driver disengagement from the driving task, led to the Jan. 22, 2018, crash in Culver City, California, according to a National Transportation Safety Board brief issued Wednesday.

Highway Accident Brief 19/07 details the results of the NTSB's investigation of the crash involving a Tesla Model S P85 and a Culver City Fire Department 2006 Seagrave Fire Truck in the high-occupancy vehicle lane of southbound Interstate 405. No one was injured as a result of the crash.

The response to a collision in the northbound freeway lanes 25 minutes earlier left a California Highway Patrol vehicle parked on the left shoulder of southbound I-405 and the Culver City Fire Department truck parked diagonally across the southbound HOV lane. Emergency lights were active on both vehicles. The Tesla, which had its "Autopilot" system engaged, was traveling in the HOV lane, behind another vehicle. After the lead vehicle changed lanes to the right, the Tesla remained in the HOV lane, accelerated and struck the rear of the fire truck at a recorded speed of about 31 mph. A forward collision warning alert occurred 0.49 seconds prior to impact but the automatic emergency braking system did not engage. The driver's hands were not detected on the steering wheel during

Related News Releases

- September 04, 2019
  Driver Errors, Advanced Driver Assistance System Design, Led to Highway Crash

Related Reports

- Highway Accident Brief: Rear-End Collision Between a Car Operating with Advanced Driver Assistance Systems and a Stationary Fire Truck

Related Events

Related Investigations

- Car with automated vehicle controls crashes into fire truck

More NTSB Links

- Investigation Process
- Data & Stats
- Accident Reports
- Most Wanted List

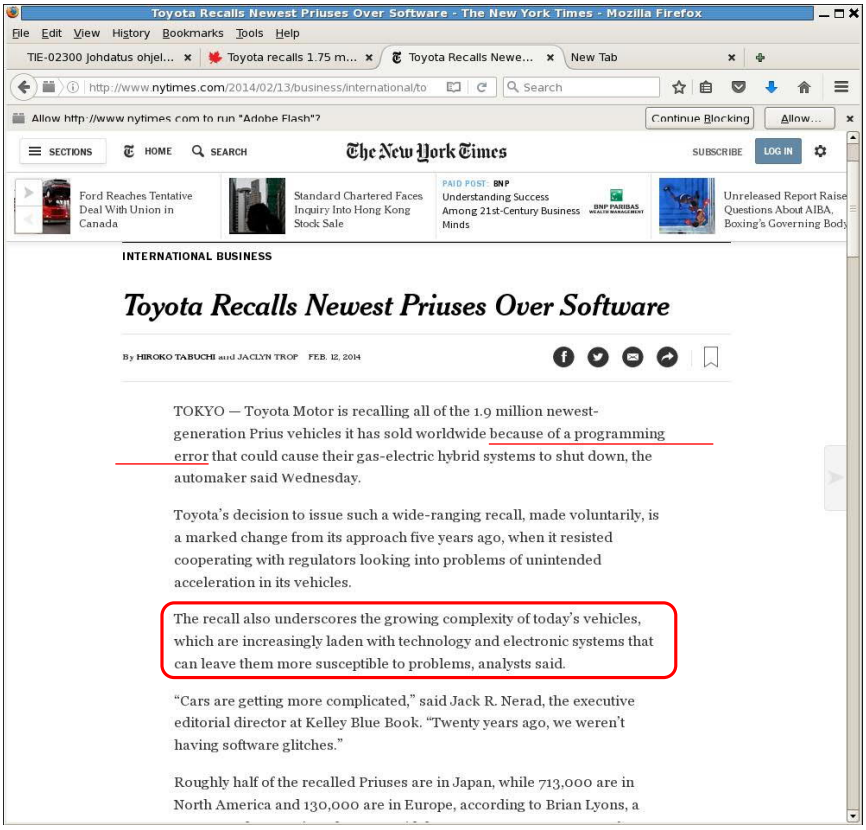
MOTHERBOARD
TECH BY VICE

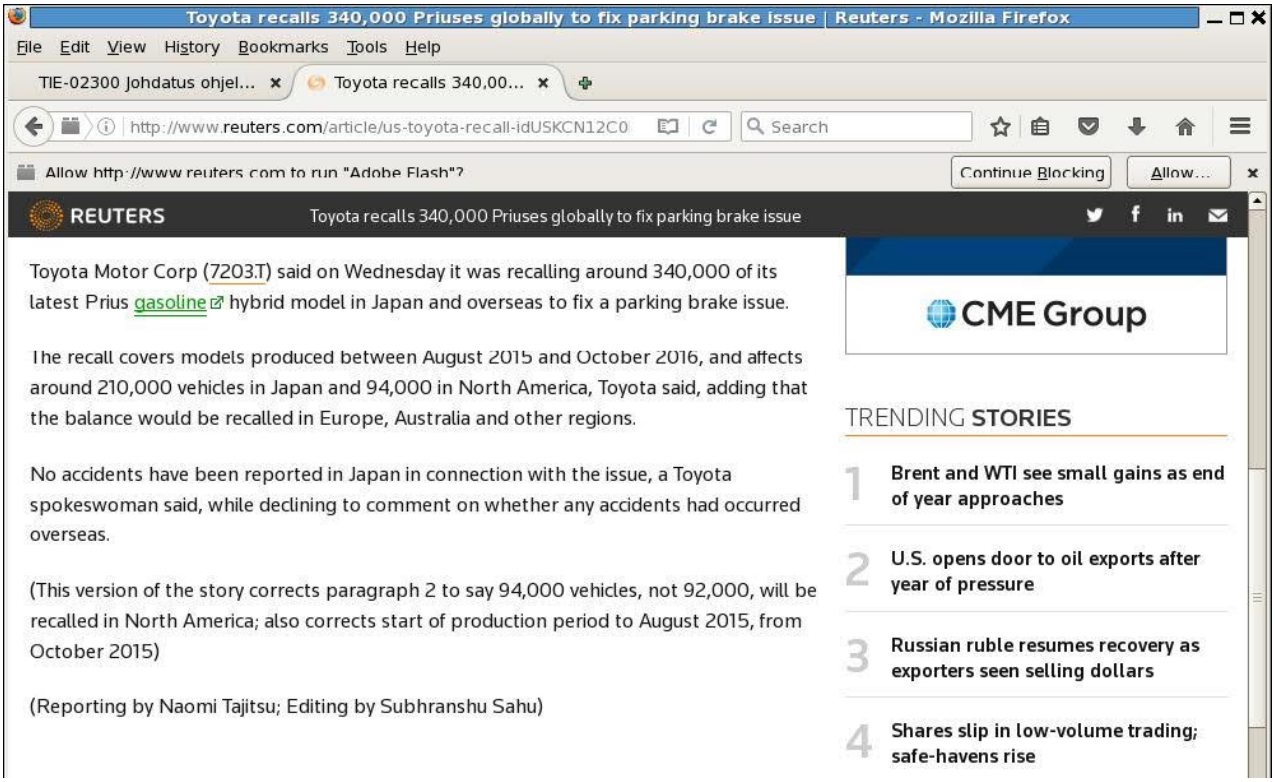
The F-35's Software Is So Buggy It Might Ground the Whole Fleet

A Government Accountability Office report confirms that the stealth fighter jets' ground-based computer system isn't working.

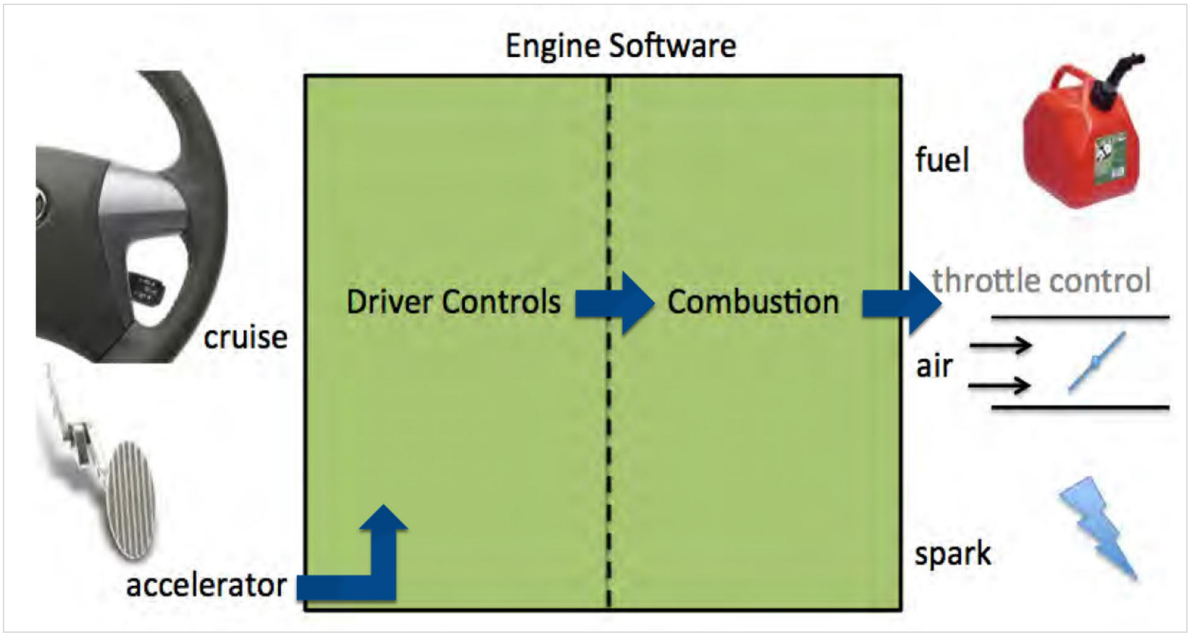
By Dan Grazier

Apr 26 2016, 9:50pm Share Tweet Snap

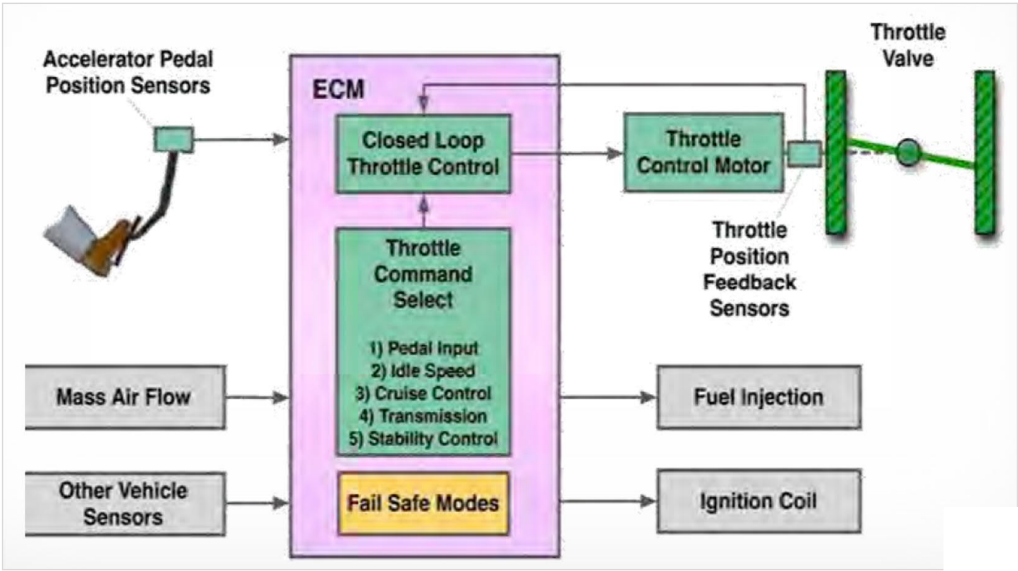

Toyota Recalls Newest Priuses Over Software - The New York Times - Mozilla Firefox
File Edit View History Bookmarks Tools Help
TIE-02300 Johdatus ohjel...
Toyota recalls 1.75 m...
Toyota Recalls Newe...
New Tab
http://www.nytimes.com/2014/02/13/business/international/to
Search
Allow http://www.nytimes.com to run "Adobe Flash"?
Continue Blocking
Allow...
SECTIONS
HOME
SEARCH
The New York Times
SUBSCRIBE
LOG IN
Ford Reaches Tentative Deal With Union in Canada
Standard Chartered Faces Inquiry Into Hong Kong Stock Sale
PAID POST: BNP
Understanding Success Among 21st-Century Business Minds
Unreleased Report Raises Questions About AIBA, Boxing's Governing Body
INTERNATIONAL BUSINESS
Toyota Recalls Newest Priuses Over Software
By HIROKO TABUCHI and JACLYN TROP FEB. 12, 2014
TOKYO — Toyota Motor is recalling all of the 1.9 million newest-generation Prius vehicles it has sold worldwide because of a programming error that could cause their gas-electric hybrid systems to shut down, the automaker said Wednesday.
Toyota's decision to issue such a wide-ranging recall, made voluntarily, is a marked change from its approach five years ago, when it resisted cooperating with regulators looking into problems of unintended acceleration in its vehicles.
The recall also underscores the growing complexity of today's vehicles, which are increasingly laden with technology and electronic systems that can leave them more susceptible to problems, analysts said.
"Cars are getting more complicated," said Jack R. Nerad, the executive editorial director at Kelley Blue Book. "Twenty years ago, we weren't having software glitches."
Roughly half of the recalled Priuses are in Japan, while 713,000 are in North America and 130,000 are in Europe, according to Brian Lyons, a


Toyota recalls 340,000 Priuses globally to fix parking brake issue | Reuters - Mozilla Firefox
File Edit View History Bookmarks Tools Help
TIE-02300 Johdatus ohjel...
Toyota recalls 340,00...
http://www.reuters.com/article/us-toyota-recall-idUSKCN12C0
Search
Allow http://www.reuters.com to run "Adobe Flash"?
Continue Blocking
Allow...
REUTERS
Toyota recalls 340,000 Priuses globally to fix parking brake issue
Toyota Motor Corp (7203.T) said on Wednesday it was recalling around 340,000 of its latest Prius gasoline hybrid model in Japan and overseas to fix a parking brake issue.
The recall covers models produced between August 2015 and October 2016, and affects around 210,000 vehicles in Japan and 94,000 in North America, Toyota said, adding that the balance would be recalled in Europe, Australia and other regions.
No accidents have been reported in Japan in connection with the issue, a Toyota spokeswoman said, while declining to comment on whether any accidents had occurred overseas.
(This version of the story corrects paragraph 2 to say 94,000 vehicles, not 92,000, will be recalled in North America; also corrects start of production period to August 2015, from October 2015)
(Reporting by Naomi Tajitsu; Editing by Subhranshu Sahu)
CME Group
TRENDING STORIES
1 Brent and WTI see small gains as end of year approaches
2 U.S. opens door to oil exports after year of pressure
3 Russian ruble resumes recovery as exporters seen selling dollars
4 Shares slip in low-volume trading; safe-havens rise

(http://www.safetyresearch.net/Library/BarrSlides_FINAL_SCRUBBED.pdf)



Same source (Toyota break case)



ECM = Engine Control Module

Joel Stocksdale

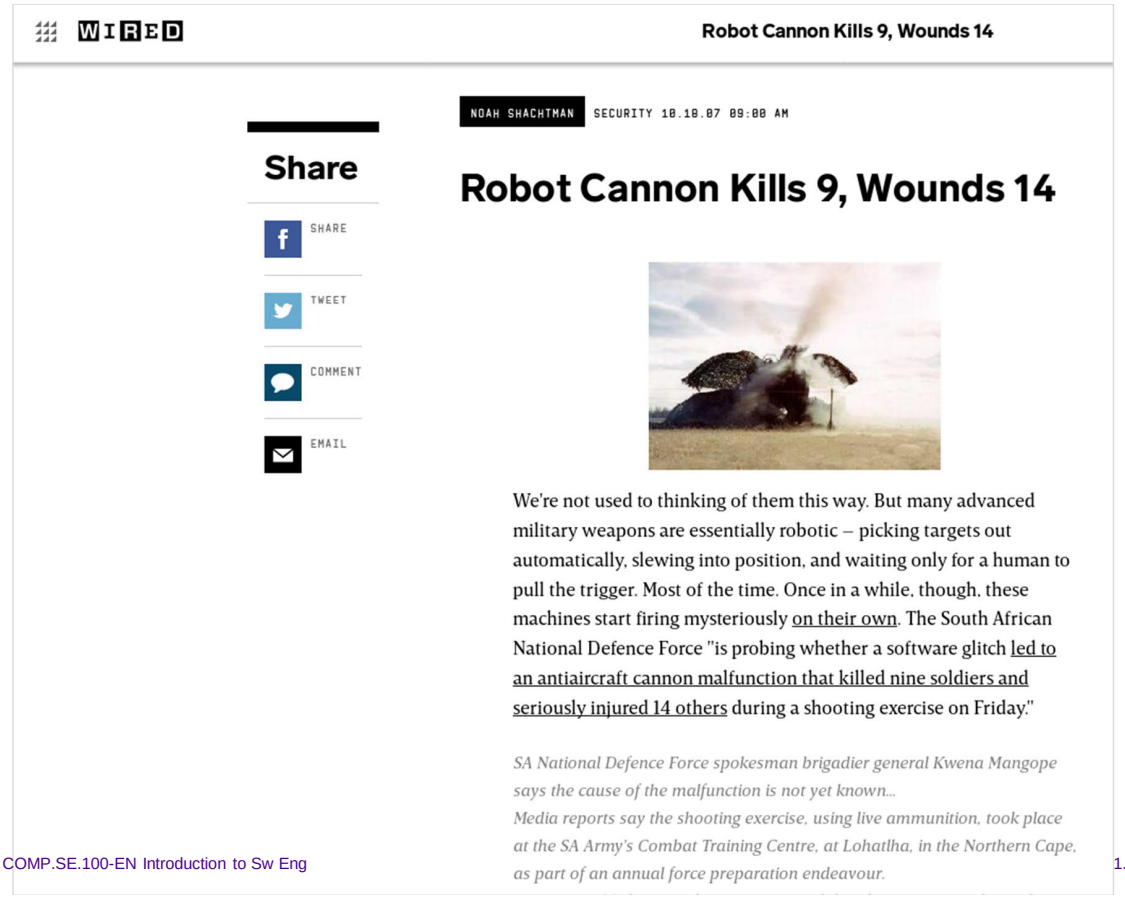
The Basics: General Motors is recalling 3.64 million vehicles across its lineup for an airbag-related issue. The recall covers the 2014-2015 Buick LaCrosse, Chevrolet SS, and Spark EV; 2014-2017 Chevrolet Corvette, Trax, Caprice PPV, Silverado 1500, Buick Encore, and GMC Sierra 1500; and 2015-2017 Chevrolet Tahoe, Suburban, Silverado HD, GMC Yukon, Yukon XL, Sierra HD, Cadillac Escalade, and Escalade ESV.

The Problem: Affected vehicles have a sensing and diagnostic module that controls the airbags and seat-belt pretensioners. The software it uses has a defect that can prompt the module to run a diagnostic test under specific driving conditions, which will also deactivate the front airbags and pretensioners. This means that it would be possible for those safety systems to not activate in a crash, potentially leading to injury or death.

Injuries/Deaths: General Motors began an investigation that led to the recall after a 2014 Silverado was involved in a crash in which the airbags did not deploy. No information was given as to injuries or deaths.

The Fix: Owners can bring their vehicles to a local General Motors dealer where a software update will be installed to fix the issue. The fix will be free of charge.

If you own one: General Motors will contact owners of affected cars, and owners can check whether their vehicles are affected by visiting entering their vehicle identification numbers at either the GM Owner Center website or the NHTSA website. Owners can then schedule a time to have the update installed.



NOAH SHACHTMAN SECURITY 10.10.07 08:59 AM

Video: Robo-Weapon's Scary Twist (Updated)

The tragedy in South Africa that killed nine soldiers isn't the first time a robotic weapon has spun out of control. Here's a video I obtained a few years back, showing a XM-151 XM-101 Common Remotely Operated Weapons Station connected to an Apache chaingun, emptying its magazine of .50-caliber bullets 30 mm high explosive rounds – and then turning towards the camera, looking for new targets to nail. I'm told – but cannot confirm – that this footage was shot during a demonstration for VIPs, and that several members of Congress would've been in serious jeopardy, had the weapon not run out of ammo.

www.youtube.com/watch?v=u7poF0M7H5M

UPDATE: I've changed the details above, thanks to info from KS, who "was there when it happened, and I was lying flat on the ground together with US Army officers."

Users seem to be nowadays afraid of searching information from company information systems, as the searches are logged (at least in health care and law enforcement), to prevent unneeded use.

Tampereen yliopisto
Tampere University

Cyber security

Tampereen yliopisto
Tampere University

ISMS = information security management system

SFS-EN ISO/IEC 27000:2020:en

ISO/IEC 27000:2018(E)

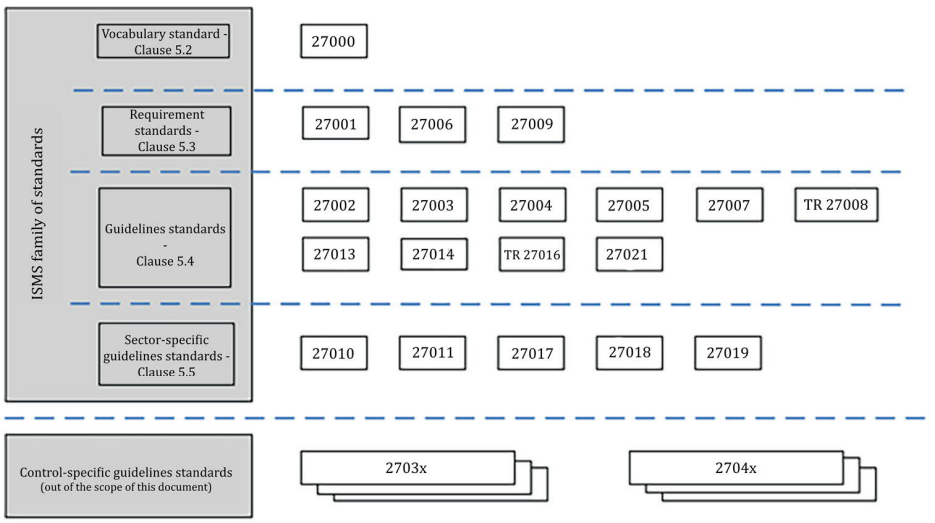
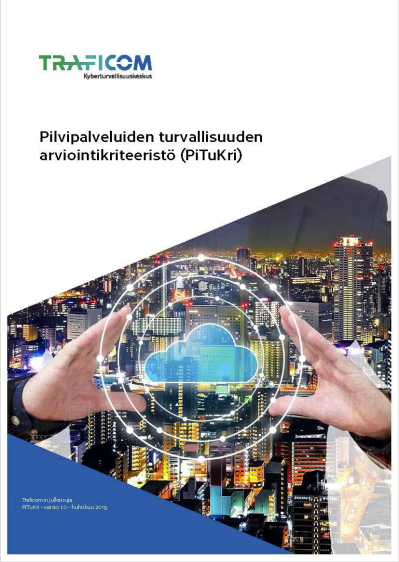


Figure 1 — ISMS family of standards relationships

Responsibilities in cloud services



	IaaS	PaaS	SaaS
	Sovellukset (Applications)	Sovellukset (Applications)	Sovellukset (Applications)
	Tieto (Data)	Tieto (Data)	Tieto (Data)
	Suorituspalvelu (Runtime)	Suorituspalvelu (Runtime)	Suorituspalvelu (Runtime)
	Väliohjelmisto (Middleware)	Väliohjelmisto (Middleware)	Väliohjelmisto (Middleware)
Asiakas (Customer/tenant)	Käyttöjärjestelmä (Operating system)	Käyttöjärjestelmä (Operating system)	Käyttöjärjestelmä (Operating system)
Palvelutarjoja (Provider)	Virtualisointi (Virtualisation)	Virtualisointi (Virtualisation)	Virtualisointi (Virtualisation)
	Palvelimet (Servers)	Palvelimet (Servers)	Palvelimet (Servers)
	Tallenuskapasiteetti (Storage)	Tallenuskapasiteetti (Storage)	Tallenuskapasiteetti (Storage)
	Verkko (Networking)	Verkko (Networking)	Verkko (Networking)
	Palvelinkeskus (Facility)	Palvelinkeskus (Facility)	Palvelinkeskus (Facility)

Kuva 1. Tyypillinen vastuujakomalli

[8] National Institute of Standards and Technology (NIST). 2011. Special Publication 800-145: The NIST Definition of Cloud Computing. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

InfoSec Maturity Model

Reactive ⟷ **Proactive**

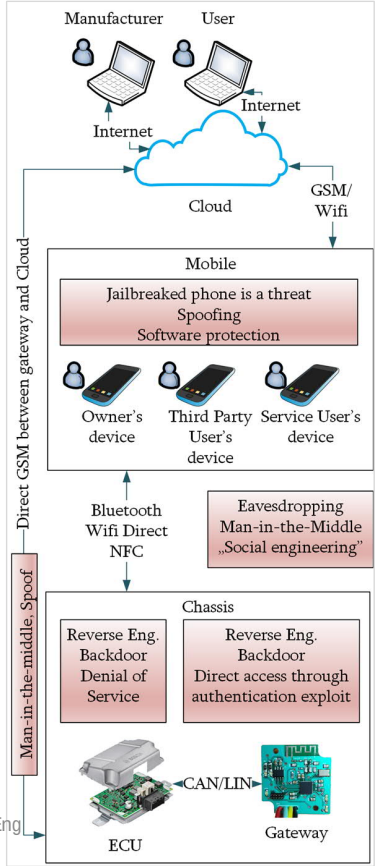
Blocking & Tackling	Compliance Driven	Risk-Based Approach

• Lack of Executive support
• Underfunded
• Understaffed
• Lack of metrics for reporting
• Set up for failure
	• Control-based security approach
	• Align to mandatory regulations
	  • EU/PII Data protection
	  • FFIEC
	  • HIPAA
	  • ISO 2700x
	  • PCI
	  • NCUA
		• Multi-layered security and risk-based approach
		• Using behavior analytics and evaluating new technologies frequently
		• Linking events across multiple disciplines

[securityboulevard.com/]

Security issues

[https://www.researchgate.net/publication/281447339_Security_issues_and_vulnerabilities_in_connected_car_systems]



Tampereen yliopisto
Tampere University

SFS Standardi

SFS-EN ISO/IEC 27000:2020

Vahvistettu 2020-02-28

1 (1 + 4 + 34)

2. painos

Korvaa: SFS-EN ISO/IEC 27000:2017 1. painos, SFS-EN ISO/IEC 27000:2017:en 1. painos

2nd edition

Replaces: SFS-EN ISO/IEC 27000:2017 1st edition, SFS-EN ISO/IEC 27000:2017:en 1st edition

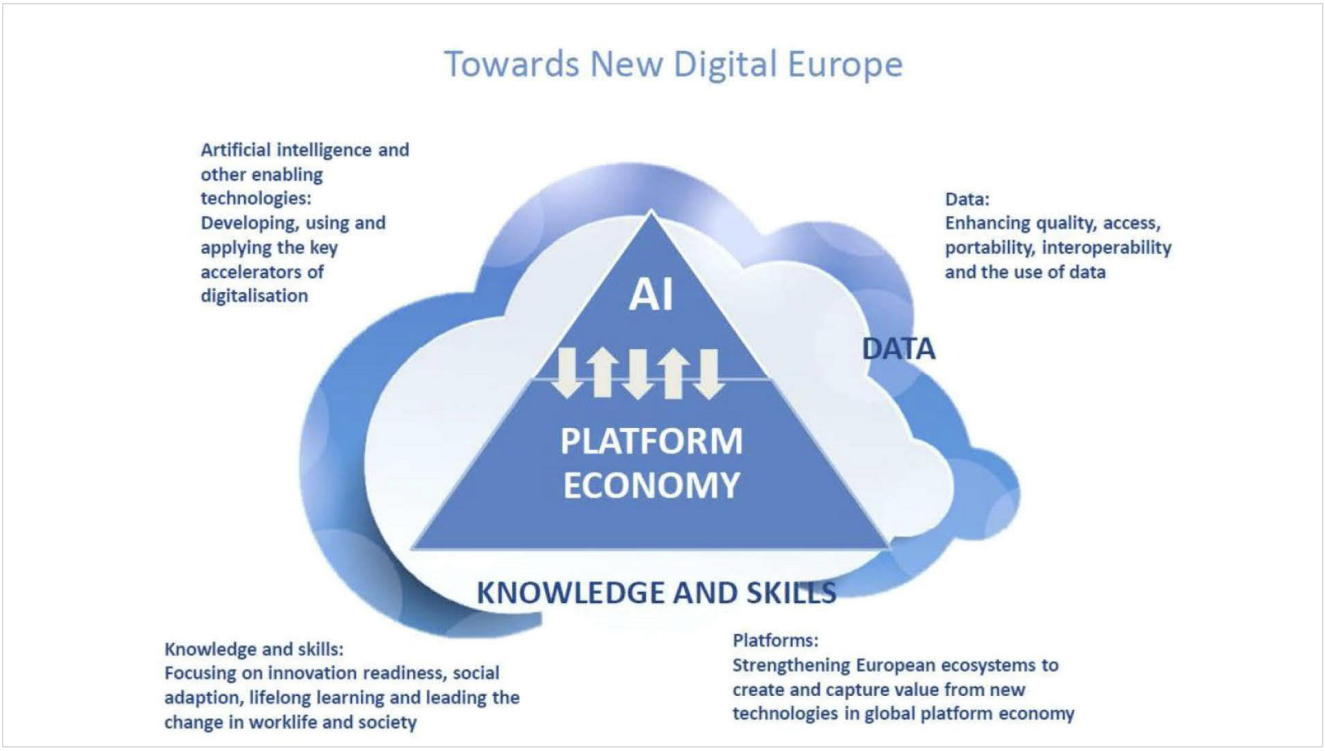
Tämä standardi on vahvistettu englanninkielisenä.	This standard is approved in English.

Information technology. Security techniques. Information security management systems. Overview and vocabulary (ISO/IEC 27000:2018)

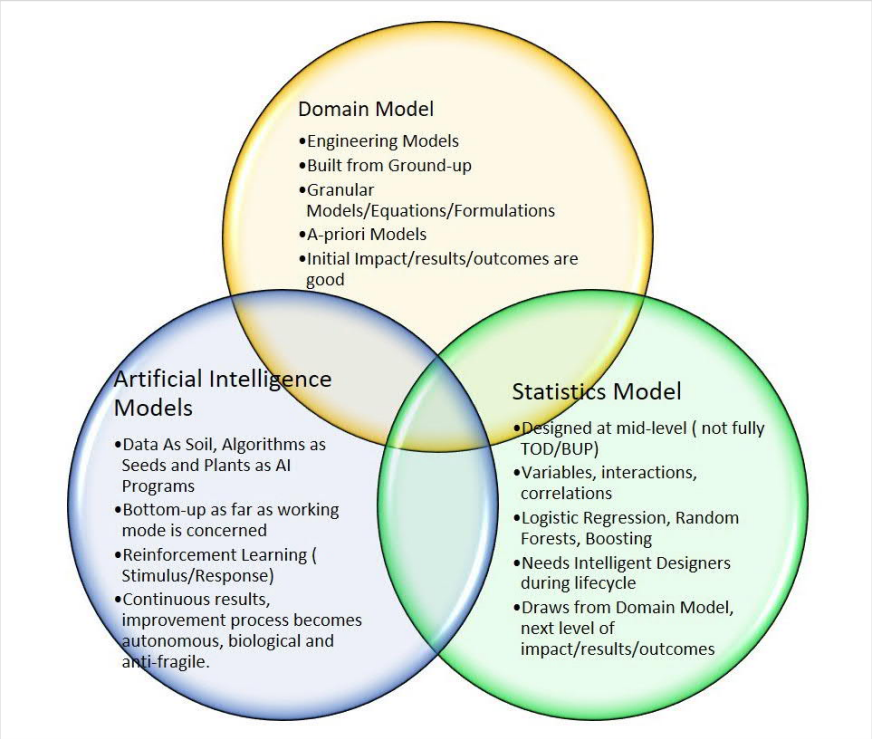
Tampereen yliopisto
Tampere University

AI, artifical intelligence

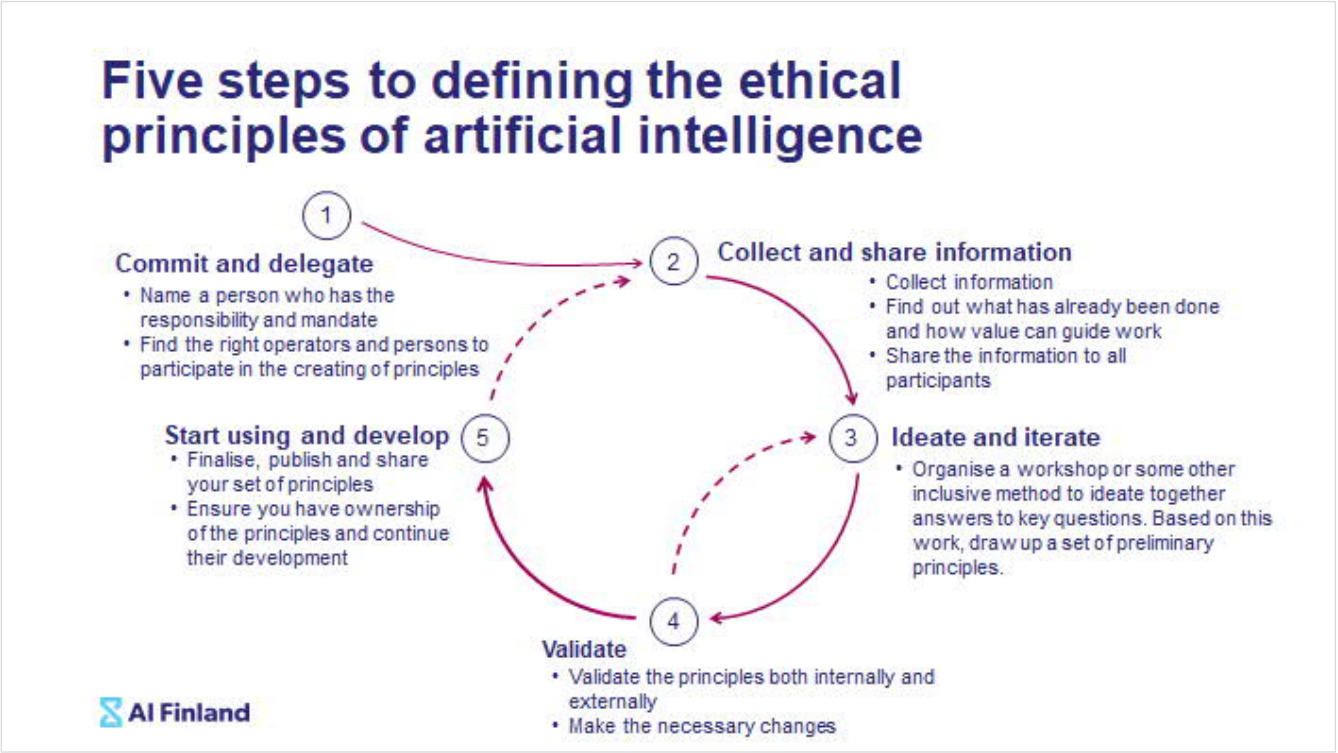
Tampereen yliopisto
Tampere University

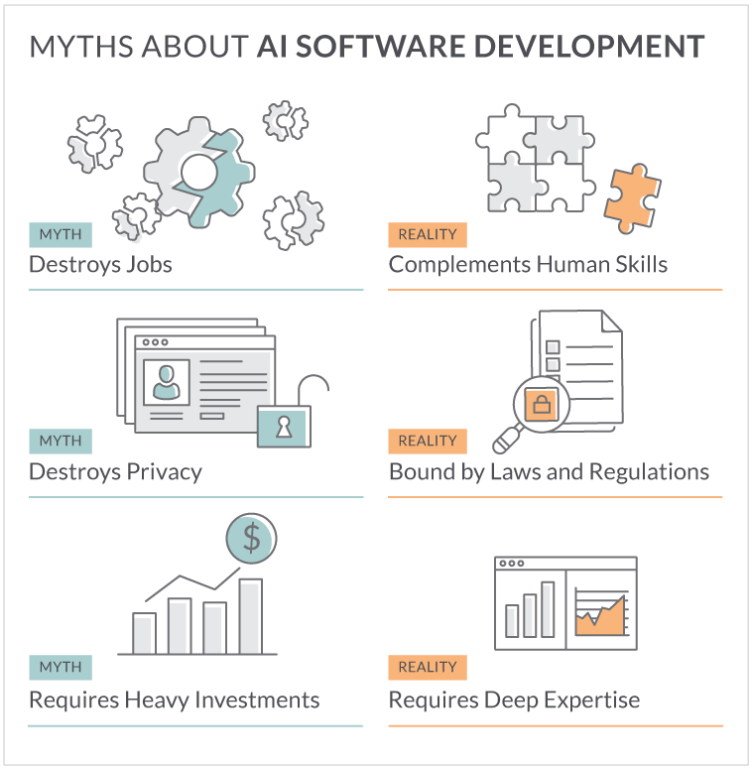


[Pekka Sivonen: FINLAND AND AI INTHE ERA OF PLATFORM ECONOMY, 2018]

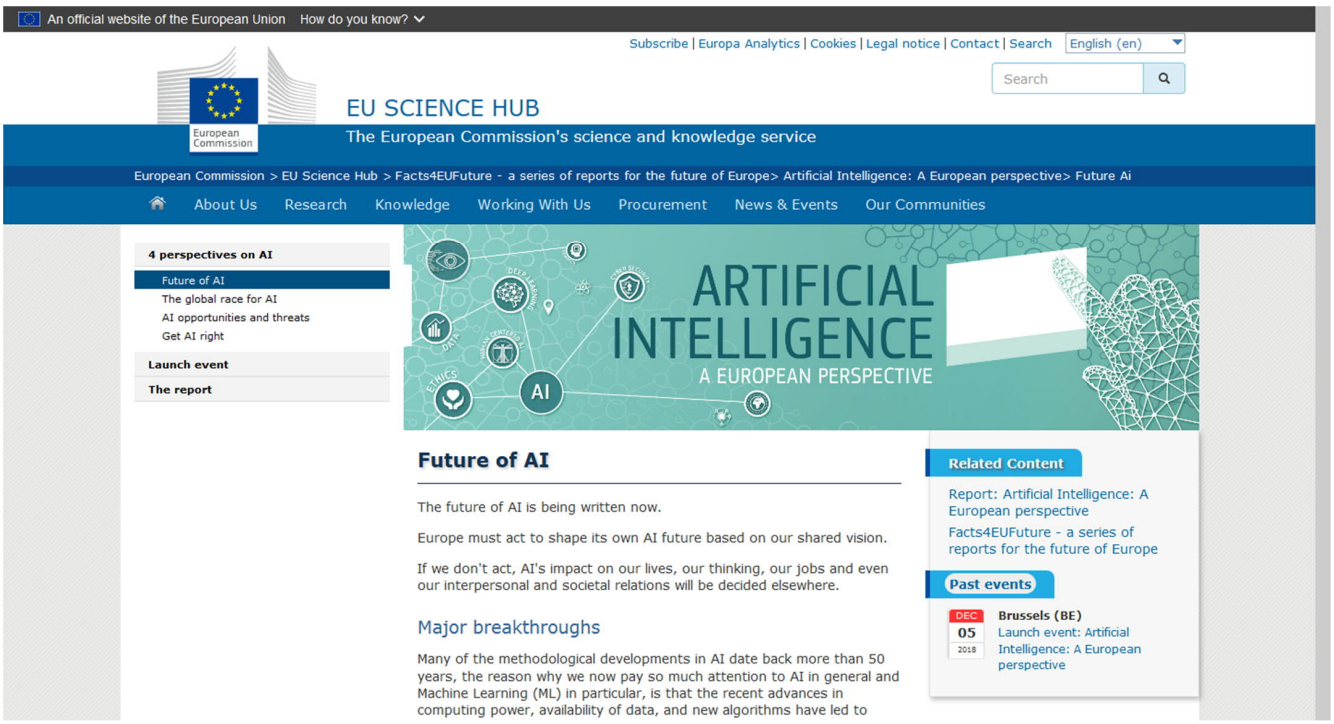


[https://medium.com/@akashmavle/domain-models-statistical-models-and-ai-ml-models-6314f03497d5]





[https://www.experfy.com/blog/ai-software-development-dispelling-the-most-common-myths]



Amazon's AI

"Everyone wanted this holy grail," one of the people said. "They literally wanted it to be an engine where I'm going to give you 100 resumes, it will spit out the top five, and we'll hire those."

But by 2015, the company realized its new system was not rating candidates for software developer jobs and other technical posts in a gender-neutral way.

That is because Amazon's computer models were trained to vet applicants by observing patterns in resumes submitted to the company over a 10-year period. Most came from men, a reflection of male dominance across the tech industry.

In effect, Amazon's system taught itself that male candidates were preferable. It penalized resumes that included the word "women's," as in "women's chess club captain." And it downgraded graduates of two all-women's colleges, according to people familiar with the matter. They did not specify the names of the schools.

Amazon edited the programs to make them neutral to these particular terms. But that was no guarantee that the machines would not devise other ways of sorting candidates that could prove discriminatory, the people said.

The Seattle company ultimately disbanded the team by the start of last year because executives lost hope for the project, according to the people, who spoke on condition of anonymity.

HYPE

Microsoft's disastrous Tay experiment shows the hidden dangers of AI

By John West • April 2, 2016

Microsoft's disastrous chatbot Tay was meant to be a clever experiment in artificial intelligence and machine learning. The bot would speak like millennials, learning from the people it interacted with on Twitter and the messaging apps Kik and GroupMe. But it took less than 24 hours for Tay's cheery greeting of "Humans are super cool!" to morph into the decidedly less bubbly "Hitler was right." Microsoft quickly took the bot offline for "some adjustments." Upon seeing what their code had wrought, one wonders if those Microsoft engineers had the words of J. Robert Oppenheimer ringing in their ears: "Now I am become death, the destroyer of worlds."

It's true that sometimes, humans were teaching Tay to hate. Daniel Victor at The New York Times writes: "Users commanded the bot to repeat their own statements, and the bot dutifully obliged."

[https://qz.com/653084/microsofts-disastrous-tay-experiment-shows-the-hidden-dangers-of-ai/]

BBC Home | News | Sport | Radio | TV | Weather | Languages | Search

[an error occurred while processing this directive]

Low graphics | Accessibility help

BBC NEWS — Watch One-Minute World News — News services: Your news when you want it

News Front Page
Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
England
Northern Ireland
Scotland
Wales
UK Politics

Last Updated: Tuesday, 25 November, 2003, 05:39 GMT

E-mail this to a friend Printable version

Computer problems hit CSA payouts

Thousands of single parents have been left short of money because of problems with a multi-million pound computer system intended to speed up payments.

Since the Child Support Agency system was introduced in April only one third of the 152,000 applications for child maintenance have been processed.

Thousands of parents are waiting for payments

SEE ALSO:

- Delayed CSA system to go online
  27 Jan 03 | Politics
- New CSA computer using 'old tricks'
  03 Oct 02 | Politics
- Computer overspend blow for CSA
  13 Aug 02 | UK News
- Child Support Agency reforms delayed
  20 Mar 02 | Politics
- CSA pays back 'wrong' dad
  17 Oct 01 | Wales

https://www.ibo.org/news/news-about-the-ib/update-m20-dp-cp-results/

17 August 2020

IB update on May 2020 Diploma Programme and Career-related Programme results

Putting the well-being of IB students and educators first, in March, we cancelled the written examinations for DP and CP.

Following the decision to cancel the examinations, our assessment specialists in collaboration with independent education experts worked to develop a reliable and valid assessment model to award grades. The awarding model used student coursework, school predicted grades and school context for final grade award. The school context was not based on previous cohorts' performance, but instead the relationship between predicted grade accuracy, performance in coursework versus examination components and final outcomes.

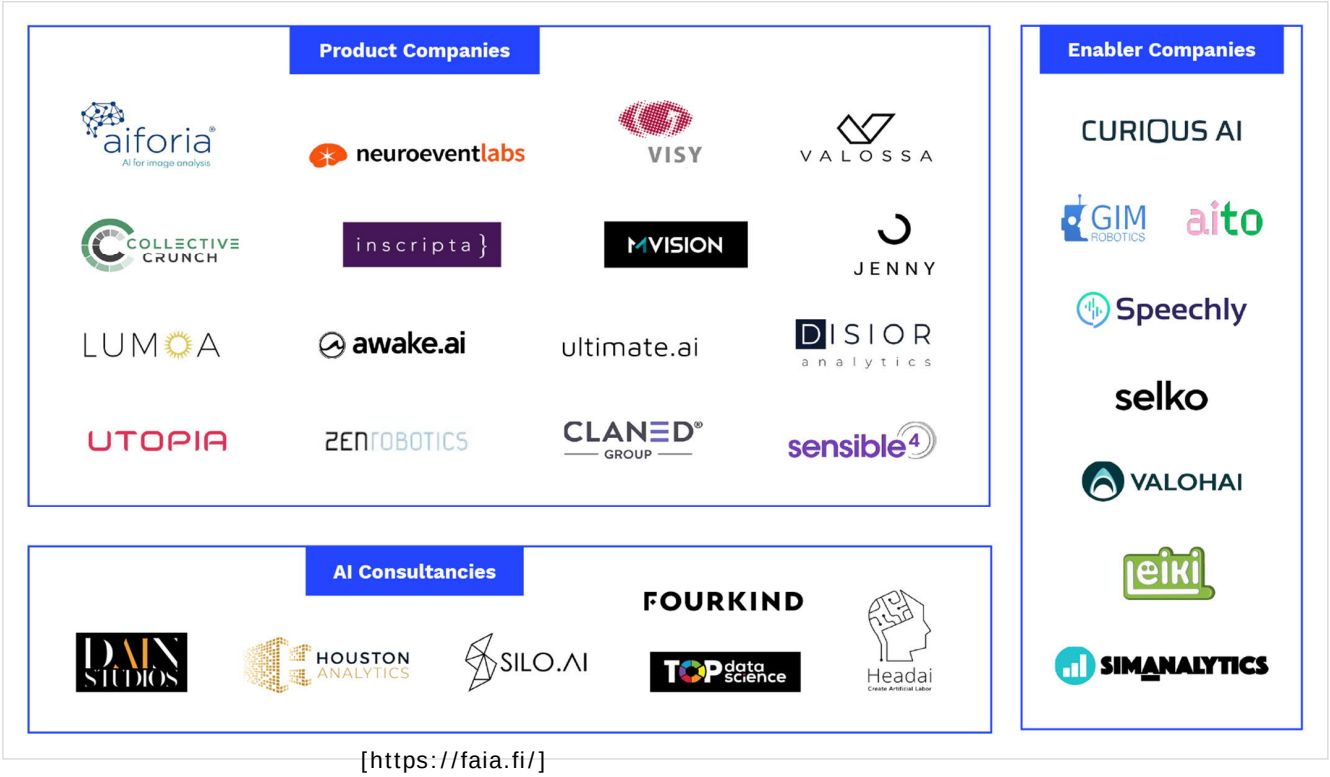
Out of the 3,020 schools receiving results in the May 2020 session, around 700 schools submitted a review request on behalf of their students.

Schools marked and submitted internal assessments (IAs) which were then marked by trained IB examiners. The IA grade is therefore a validated data point in determining the final grade. Schools also provided predicted grades (PGs) based on their judgement as to what candidates would have received had they sat the examinations.

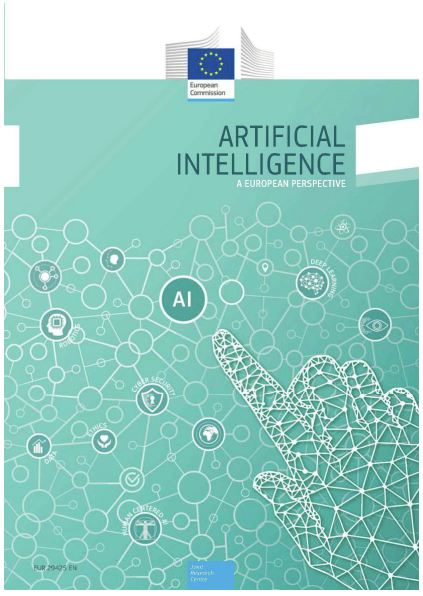
Students' subject **final grade results will be adjusted** to be equal to the internal assessment (IA) result when the predicted grade was only one grade less, equal to or greater than the IA grade.

Who is responsible ?

Finland's AI landscape



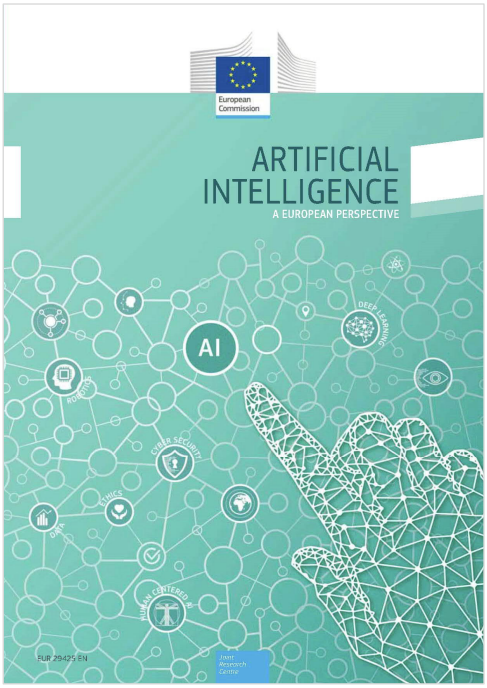
[https://faia.fi/]



Data is the lifeline of AI.

Europe is data rich but the economics and legal framework create complex challenges.

Opening access to data and building interactions among participants is key to succeeding.



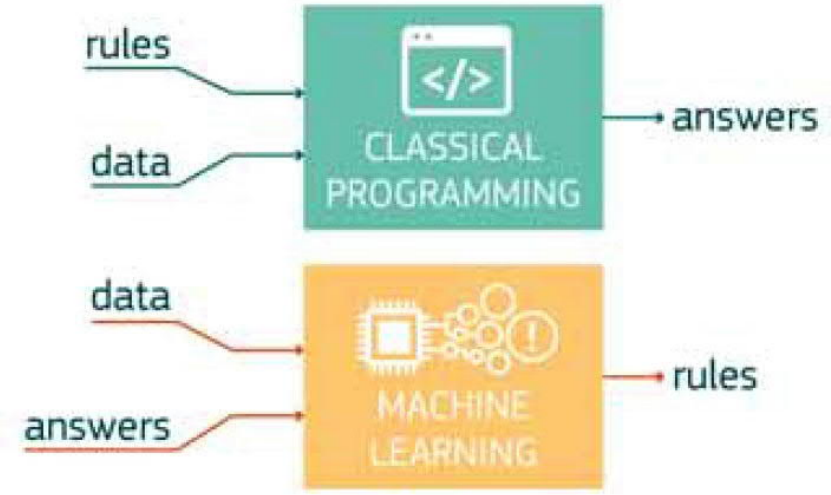


Figure 1. Paradigm shift in ML[2]
Source: after Chollet, 2017

BOX 4. Some examples of AI developments in the USA

Google is an AI world leader. It researches and develops AI products and services, has developed its own AI chip, the Tensor Processing Unit (TPU), and created TensorFlow, one of the most widely used open source AI/ML libraries. DL algorithms are already a considerable part of several Google products including recommendation systems, Android, Gmail, Maps, and translation services. The number of DL directories Google uses has gone from zero in 2012 to 4 000 in 2016.[14] In May 2018, Google announced Google duplex, an AI system for accomplishing real-world tasks over the phone. Google, and other large operators, also use AI extensively to manage their data centres and save energy.[15]

Facebook is the biggest social media company in the world. It has established a Facebook AI research (FAIR) team which is one of the most advanced, with several labs around the world. Facebook supports the open source AI libraries PyTorch and Caffe2 that compete with Google TensorFlow. The company has also released plans to develop its own AI chips.

Amazon uses AI extensively in its recommendation systems and logistics. It also offers both consumer and business-oriented AI products and services. Amazon Echo brings AI into the home through the intelligent voice server, Alexa. For business, Amazon AI empowers fake reviews detection, chatbots, product recommendations, big data management, etc. Amazon Web Services are probably the biggest cloud system in the world providing infrastructure as a service to over 1 million users.[16]

Microsoft has also been investing heavily in AI and infrastructure as a service. Microsoft Azure is one of the three biggest cloud (and AI) providers together with Google Cloud and Amazon web services. Cortana is the Microsoft virtual assistant that competes with Amazon Alexa, Apple Siri, Google Duplex and others.

Apple is the most valuable company in the world worth over US$ 1 trillion. Apple's main AI divisions are Siri team, and the Core ML team. Siri's team focus is on NLP and computer vision, both of which are necessary to power voice assistant features and new, more cutting-edge technology such as augmented reality apps that rely on object recognition. Core ML is the ML Application Programming Interface (API) that Apple launched last year to help AI tasks and AI-focused apps and services from third-party developers run more efficiently on Apple devices.

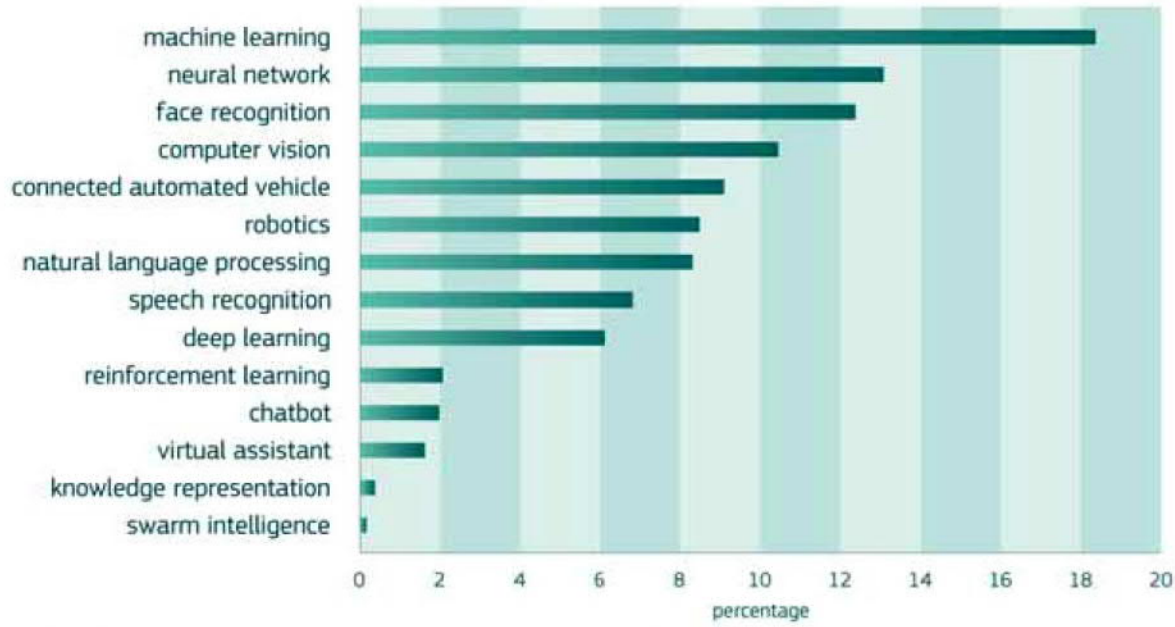
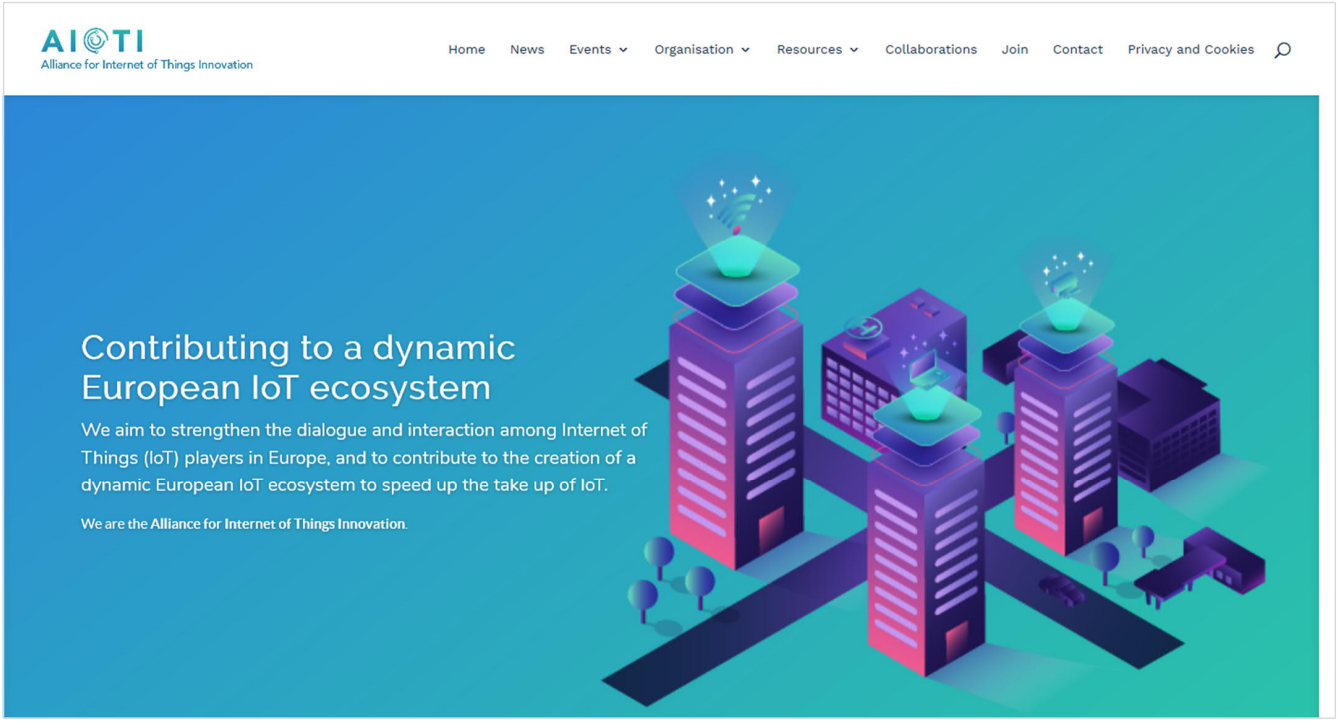


Figure 2. Percentage of AI players detected by the most representative keywords (2000-18)
Note: The conceptual proximity of some of the keywords, and the overlap between techniques and applications, may result in the same player been detected more than once.



[https://aioti.eu/]

BOX 9. Summary data strategy from the Mission Villani Report (2018)

1. Encourage companies to pool and share their data

The government must encourage the creation of data commons and support an alternative data production and governance model based on reciprocity, cooperation and sharing. The goal is to boost data sharing between actors in the same sector. The government must also encourage data sharing between private actors, and assist businesses in this respect. It must organise for certain data held by private entities to be released on a case-by-case basis, and support data- and text-mining practices without delay.

2. Create data that is in the public interest

Most of the actors heard by the mission were in favour of progressively opening up access to some data sets on a case-by-case and sector-specific basis for public interest reasons. This could be in one of two ways: by making the data accessible only to the government, or by making the data more widely available, for example, to other economic actors.

3. Support the right to data portability

The right to data portability is one of the most important innovations in recent French and European texts. It will give any individual the ability to migrate from one service ecosystem to another without losing their data history. This right could be extended to all citizen-centred artificial intelligence applications. In this case, it would involve making personal data available to government authorities or researchers.

The following key measures should be introduced during the next 12 months (1/2):

Clarify the rules of how data is used, from the perspective of companies, society and users. Provide support for the use of data by means of legislation, agreements and self-regulation of industries.

Support the development of significant testbeds and international cooperation. Integrate the operations as part of the Finnish Digital Innovation Hub network.

Recognise the business potential of different types of ecosystems and the B2B market and develop solutions for using data in them.

Continue AI accelerator style operations based on the lessons learned and seek opportunities to expand the operations.

Ensure Finland's ability to secure major strategic investments in AI and RDI investments in competences.

On the basis of the experiences gained, create an extensive provision of online courses for those in working life, which would provide an opportunity for the adult population to supplement and renew their competences.

[Leading the way into the era of artificial intelligence, 2019]

Tampereen yliopisto
Tampere University

The following key measures should be introduced during the next 12 months (2/2):

Explore whether every Finn in working age could be provided with a learning voucher or account, which would create a well-functioning adult education market in Finland.

Ensure human-centric introduction of artificial intelligence and the implementation of ethical principles in the public sector through the AuroraAI project.

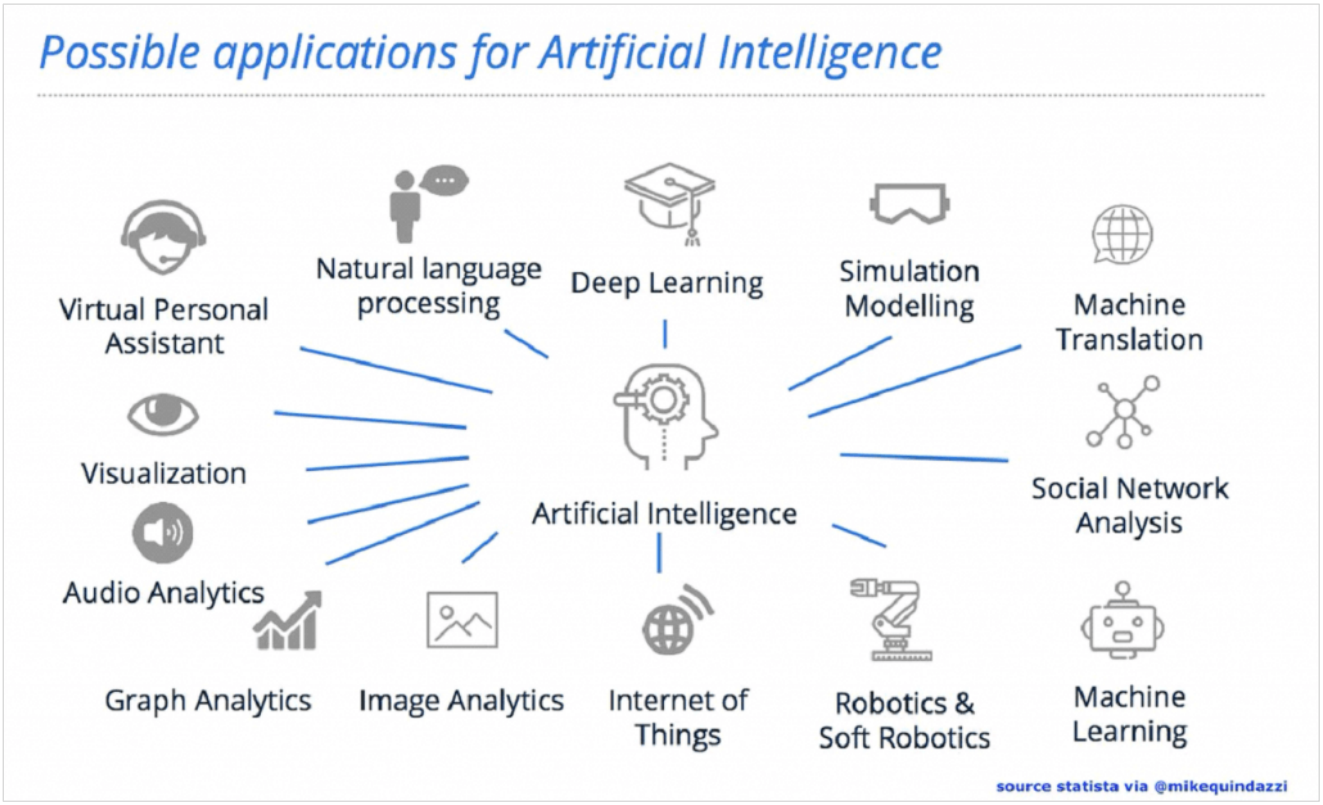
Encourage companies and public-sector actors to introduce ethical self-regulation and to share best practices.

Introduce the digital economy, founded on artificial intelligence, data and platform economy, as one of the key themes of Finland's EU Presidency.

Monitor how the implementation of the Artificial Intelligence Programme's objectives is advancing. The responsibility for the monitoring should belong to a monitoring group with representatives from both the private and public sectors or a broader cooperation forum promoting the digitalisation of business.

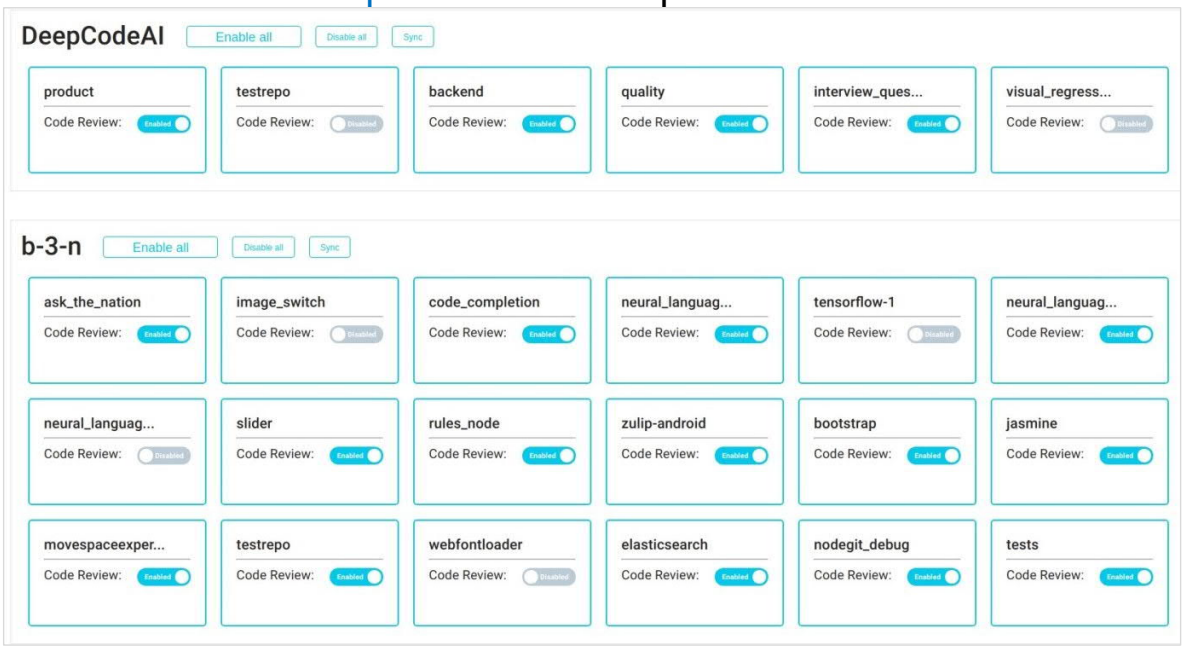
[Leading the way into the era of artificial intelligence, 2019]

Tampereen yliopisto
Tampere University



Tampereen yliopisto
Tampere University

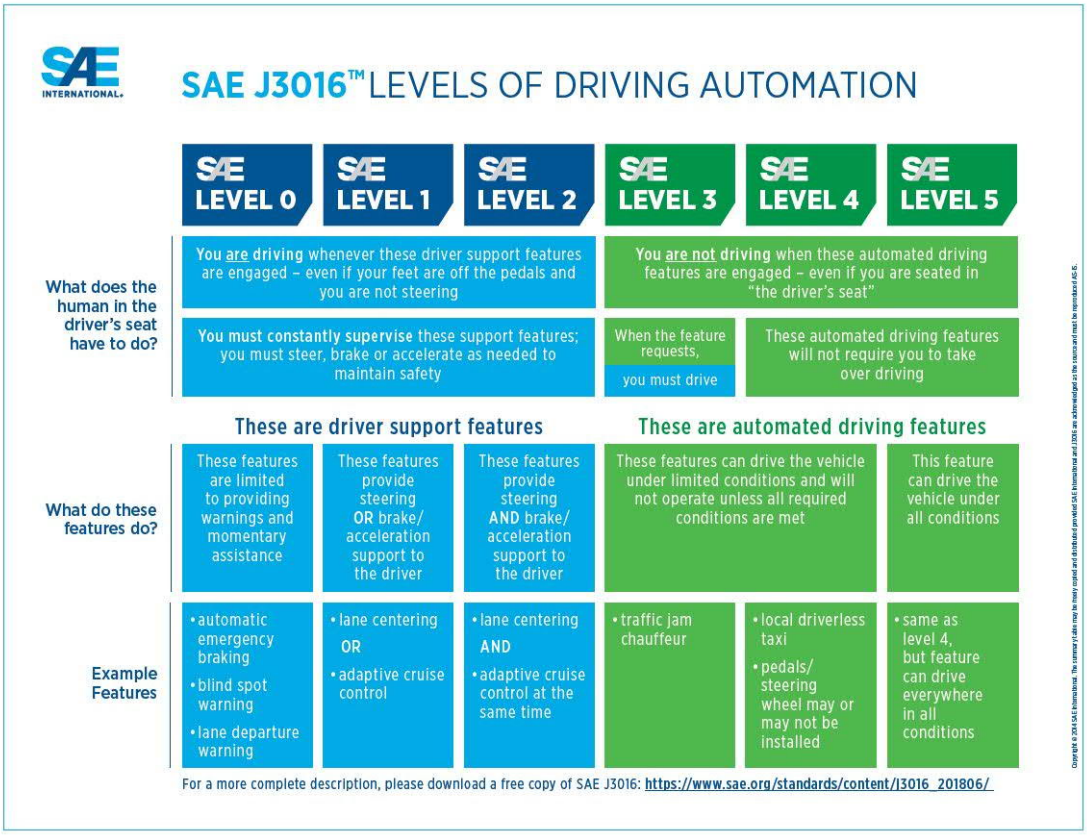
How AI Powered Tools Are Bringing Revolution to Software Development? 12 April 2019

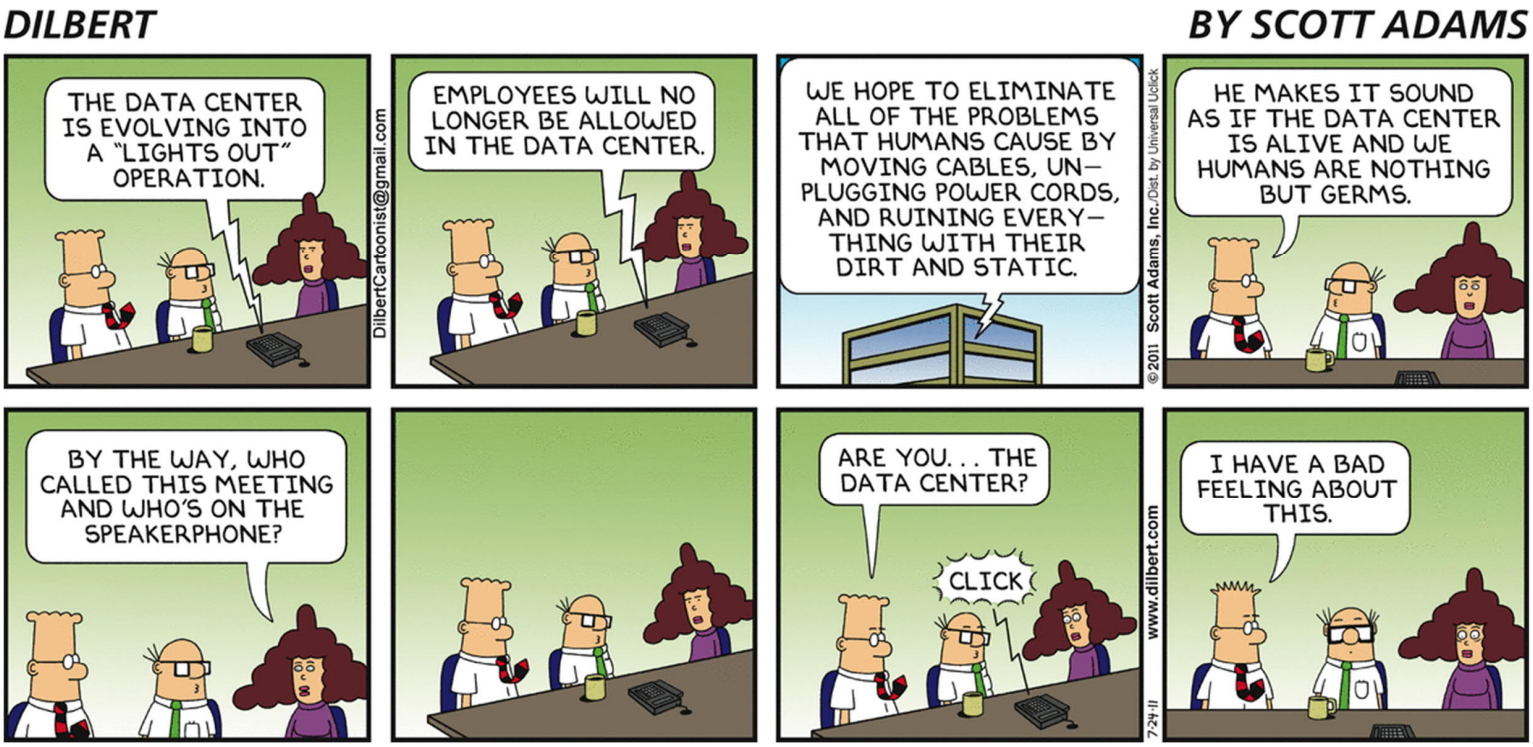


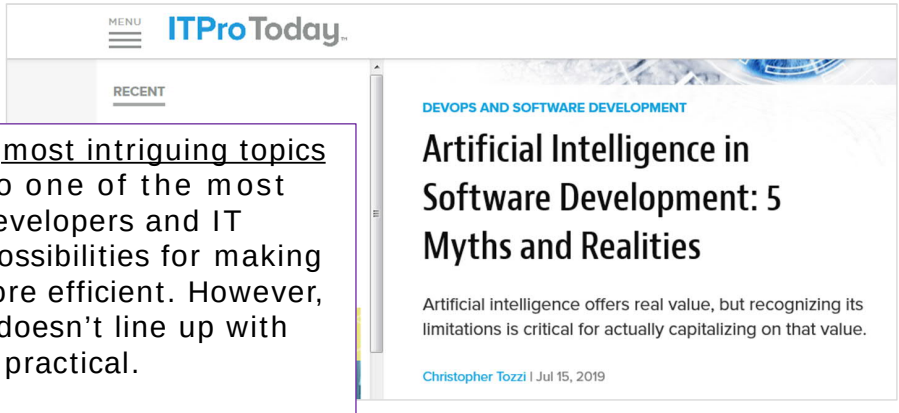
Tampereen yliopisto
Tampere University

SAE (Society of Automotive Engineers) Six levels of driving automation, 2018

[https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic]



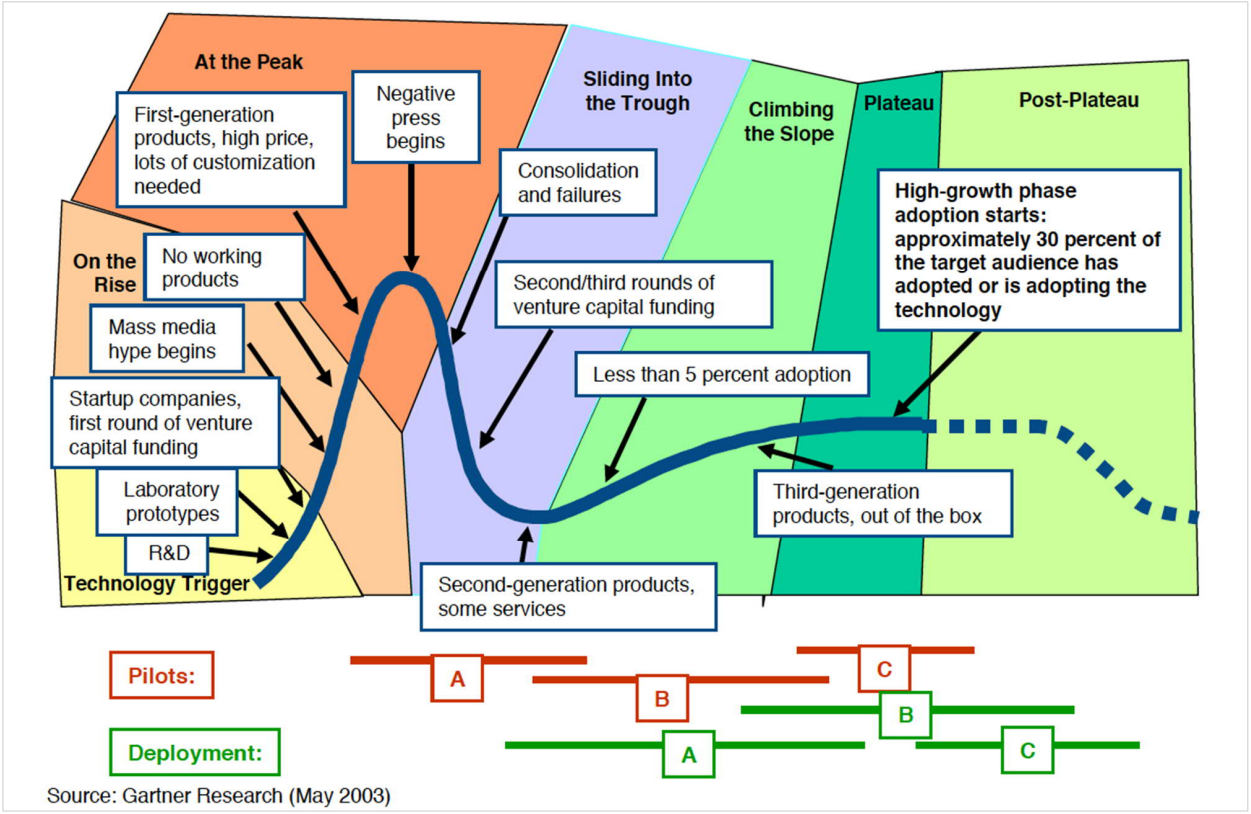




Artificial intelligence, or AI, is one of the most intriguing topics in software development today. It is also one of the most widely misunderstood. For software developers and IT teams, AI offers an array of tantalizing possibilities for making applications faster, more scalable and more efficient. However, in many cases, the hype surrounding AI doesn't line up with the reality of what is actually possible or practical.

1. Artificial inteligence is a new technology.
2. AI is smarter than humans.
3. AI will lead to smaller IT teams.
4. AI software is "set and forget."
5. AI will destroy the world.

(all are wrong assumptions)


At the Peak
Sliding Into the Trough
Climbing the Slope
Plateau
Post-Plateau
First-generation products, high price, lots of customization needed
Negative press begins
Consolidation and failures
High-growth phase adoption starts: approximately 30 percent of the target audience has adopted or is adopting the technology
On the Rise
No working products
Mass media hype begins
Second/third rounds of venture capital funding
Less than 5 percent adoption
Startup companies, first round of venture capital funding
Laboratory prototypes
R&D
Technology Trigger
Third-generation products, out of the box
Second-generation products, some services
Pilots:
Deployment:
A
B
C
B
A
C
Source: Gartner Research (May 2003)


WALLY, I NEED YOU TO HEAD UP OUR ARTIFICIAL INTELLIGENCE PROJECT.
YOU WILL HAVE NO BUDGET AND NO HOPE OF SUCCESS. I JUST LIKE SAYING WE'RE WORKING ON AI.
AND YOU'RE COMPLETELY USELESS, SO IT'S A GOOD MATCH.
I WON'T LET YOU DOWN.
Dilbert.com @ScottAdamsSays
6-20-16 © 2016 Scott Adams, Inc. /Dist. by Universal Uclick

Tampereen yliopisto
Tampere University

Bloomberg

Business

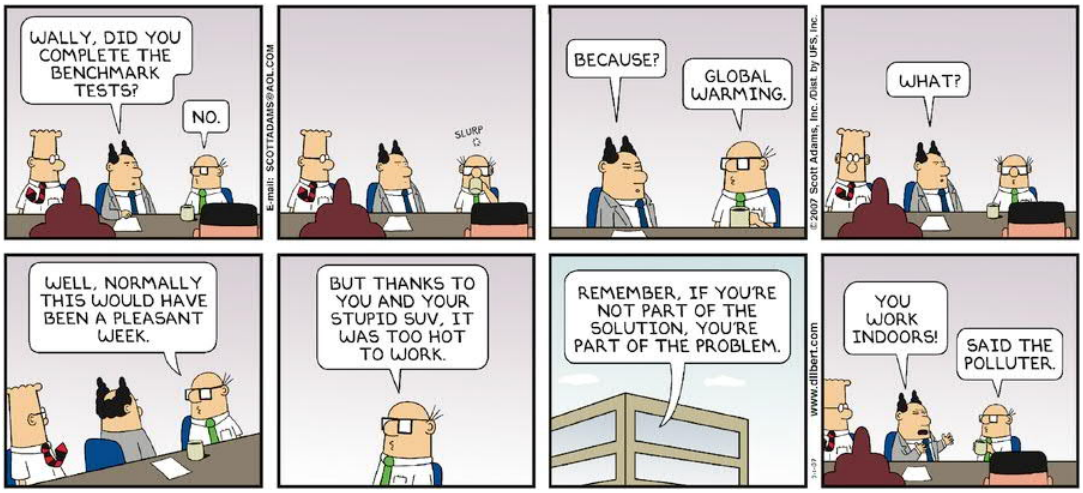
Knight Shows How to Lose $440 Million in 30 Minutes

By Matthew Philips
August 3, 2012, 1:10 AM GMT+3

Talk about a bad day. In the mother of all computer glitches, market-making firm Knight Capital Group lost $440 million in 30 minutes on Aug. 1 when its trading software went, to use the technical term, kablooey. That's four times its net income from all of 2011, and a lot more than most analysts were

Tampereen yliopisto
Tampere University

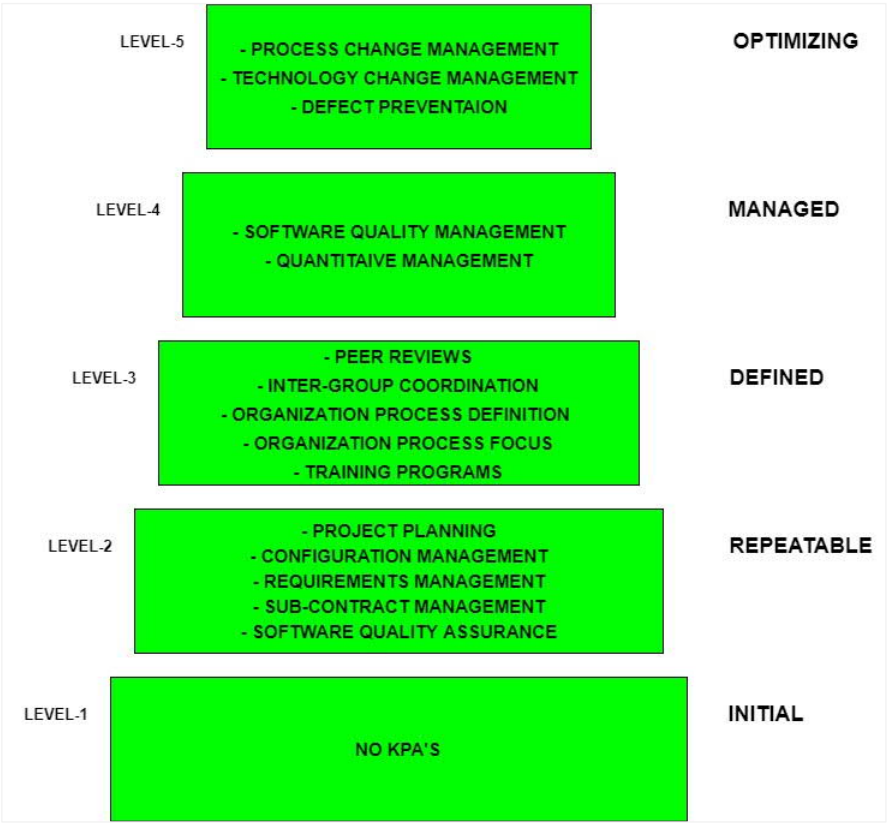
Benchmarking



Tampereen yliopisto
Tampere University

CMM /CMM*

CMM levels



[https://www.geeksforgeeks.org/software-engineering-capability-maturity-model-cmm/]

What is CMM [https://www.tutorialspoint.com/cmmi/cmmi_overview.htm]

CMM stands for Capability Maturity Model.

Focuses on elements of essential practices and processes from various bodies of knowledge.

Describes common sense, efficient, proven ways of doing business (which you should already be doing) – not a radical new approach.

CMM is a method to evaluate and measure the maturity of the software development process of an organization.

CMM measures the maturity of the software development process on a scale of 1 to 5.

CMM was originally developed for Software Development and Maintenance but later it was developed for

- Systems Engineering
- Supplier Sourcing
- Integrated Product and Process Development
- People CMM
- Software Acquisition.

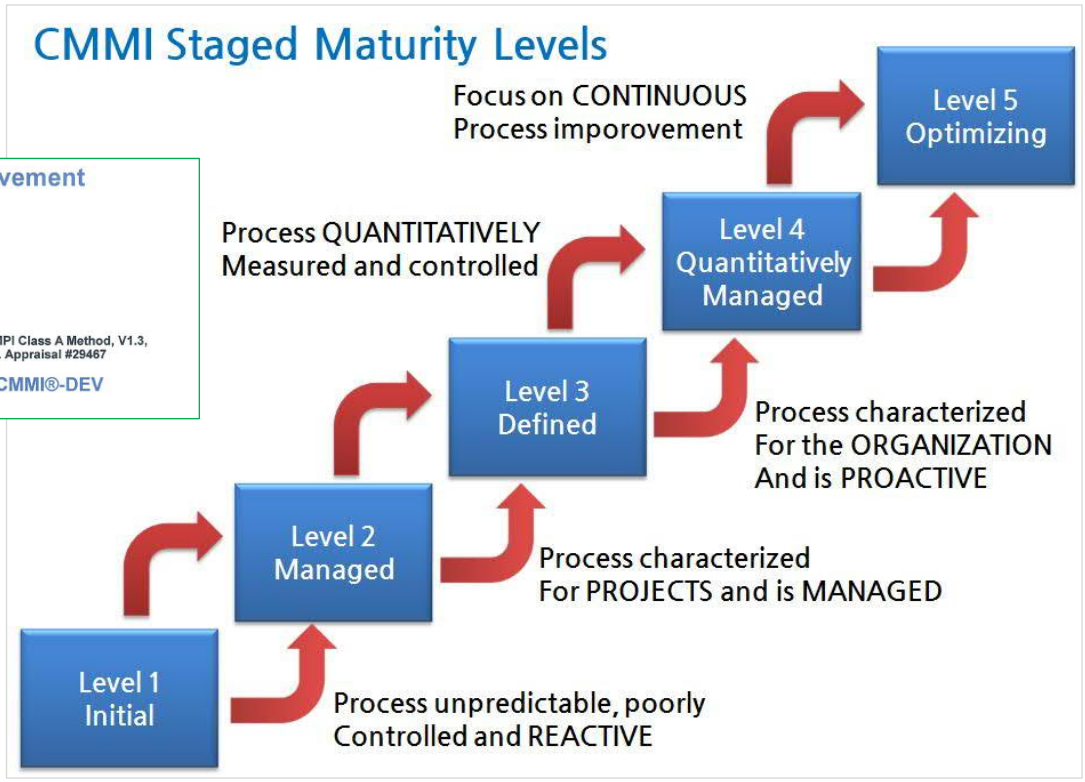
CMMI [https://www.geeksforgeeks.org/capability-maturity-model-integration-cmmi/]

Capability Maturity Model Integration (CMMI) is a successor of CMM and is a more evolved model that incorporates best components of individual disciplines of CMM like Software CMM, Systems Engineering CMM, People CMM, etc.

Since CMM is a reference model of matured practices in a specific discipline, so it becomes difficult to integrate these disciplines as per the requirements. This is why CMMI is used as it allows the integration of multiple disciplines as and when needed.

Objectives of CMMI :

- Fulfilling customer needs and expectations.
- Value creation for investors/stockholders.
- Market growth is increased.
- Improved quality of products and services.
- Enhanced reputation in Industry.



[https://iqmsglobal.com/cmmi/]

Tampereen yliopisto
Tampere University

CMMI Levels of Capability and Performance

Capability Level 0: Incomplete

Incomplete approach to meeting the intent of the Practice Area.

May or may not be meeting the intent of any practice.

Inconsistent performance.

Capability Level 1: Initial

Initial approach to meeting the intent of the Practice Area.

Not a complete set of practices to meeting the full intent of the Practice Area.

Addresses performance issues.

Tampereen yliopisto
Tampere University

CMMI Levels of Capability and Performance

Capability Level 2: Managed

Subsumes level 1 practices.

Simple, but complete set of practices that address the full intent of the Practice Area.

Does not require the use of the organizational assets.

Identifies and monitors progress towards project performance objectives.

Capability Level 3: Defined

Builds on level 2 practices.

Uses organizational standards and tailoring to address project and work characteristics.

Projects use and contribute to organization assets.

Focuses on achieving both project and organizational performance objectives.

Tampereen yliopisto
Tampere University

Certificates and certification

Tampereen yliopisto
Tampere University

AQAP use

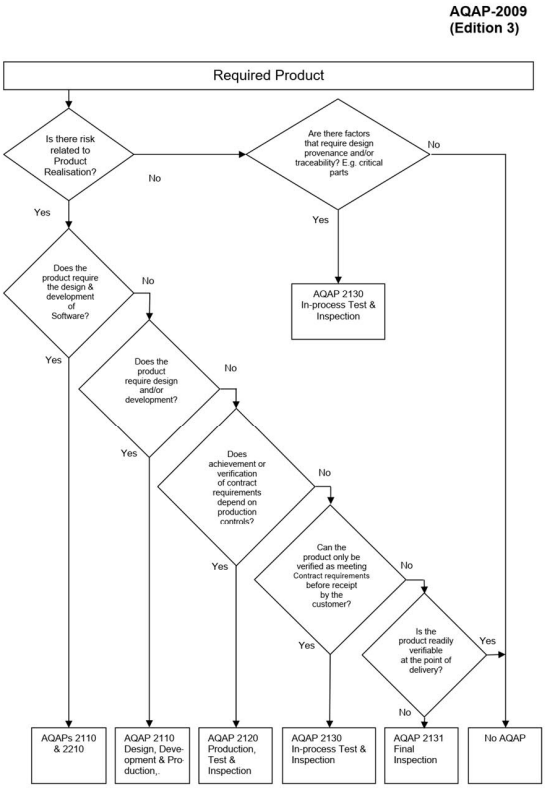
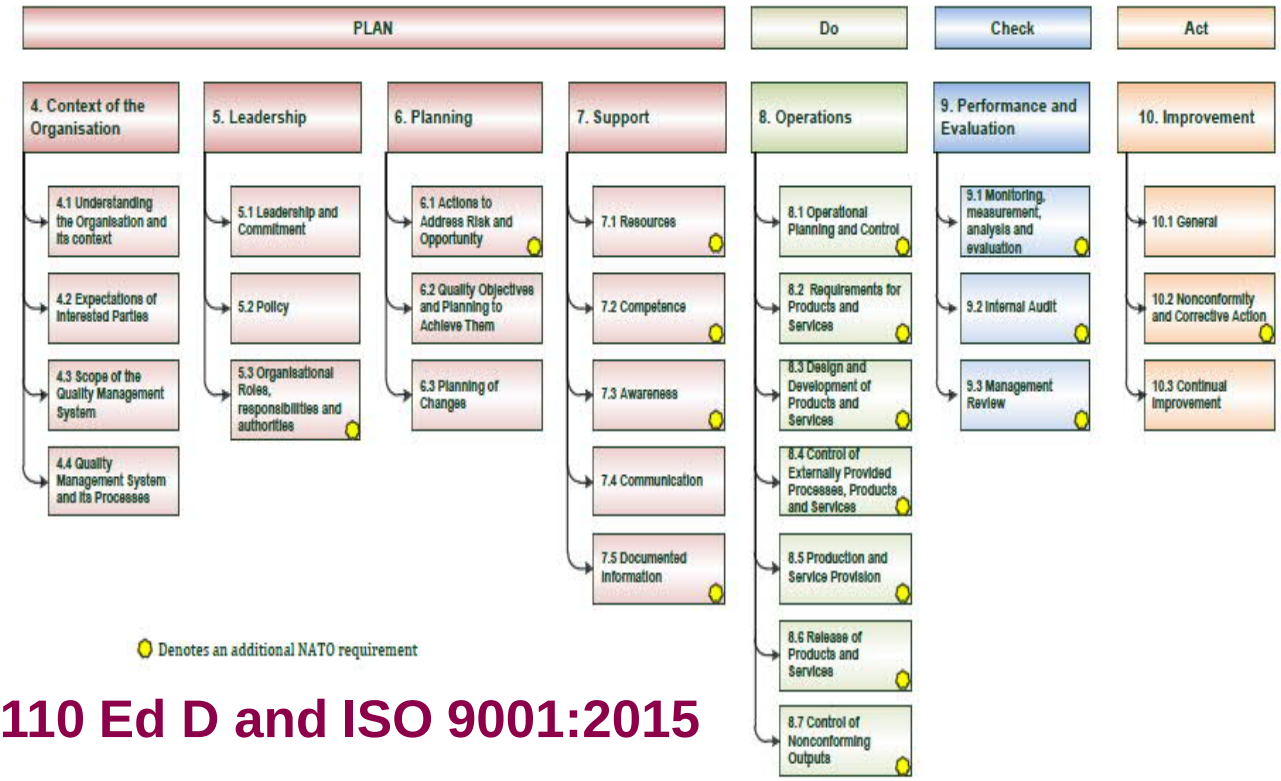


Figure 2
Possible process flow indicated



AQAP 2110 Ed D and ISO 9001:2015

Tampereen yliopisto
Tampere University